TAR for Smart People

Expanded and Updated Third Edition

How Technology-Assisted Review Works and Why It Matters for Legal Professionals

By John Tredennick
with Jeremy Pickens, Thomas C. Gricks III & Andrew Bye
“It is said that lawyers and judges went to law school because they were promised there would be no math. Their resistance to technological change is legendary. But, now, hell has frozen over. There is a book about TAR that is clear, incisive and actually fun to read. Lawyers now have no good reason to avoid understanding a technology which will transform discovery as nothing else has ever done.”

-U.S. Magistrate Judge John M. Facciola (Retired)
John Tredennick, the founder of Catalyst (now OpenText™), is a nationally known trial lawyer, e-discovery expert and former longtime litigation partner at Holland & Hart. Prior to founding Catalyst in 2000, John was a pioneer in the field of legal technology. He was editor-in-chief of the best-selling, multi-author, two-book series, *Winning With Computers: Trial Practice in the Twenty-First Century* (ABA Press 1990, 1991). At the same time, he wrote, *How to Prepare for, Take and Use a Deposition at Trial* (James Publishing 1990), which he and his co-author continued to supplement for several years. He also wrote *Lawyer’s Guide to Spreadsheets* (Glasser Publishing 2000), and *Lawyer’s Guide to Microsoft Excel 2007* (ABA Press 2009).

John is the former chair of the ABA’s Law Practice Management Section. For many years, he was editor-in-chief of the ABA’s *Law Practice Management* magazine. Over two decades, John has written scores of articles on legal technology and spoken on legal technology to audiences on four of the five continents.
Jeremy Pickens, Ph.D.

Jeremy Pickens is one of the world's leading information retrieval scientists and a pioneer in the field of collaborative exploratory search, a form of information seeking in which a group of people who share a common information need actively collaborate to achieve it. Dr. Pickens has seven patents and patents pending in the field of search and information retrieval.

Before joining OpenText, Dr. Pickens spearheaded the development of Insight Predict at Catalyst. His ongoing research and development focuses on methods for continuous learning, and the variety of real world technology-assisted review workflows that are only possible with this approach.

Dr. Pickens earned his doctoral degree at the University of Massachusetts, Amherst, Center for Intelligent Information Retrieval. He conducted his post-doctoral work at King's College, London. In addition to his OpenText responsibilities, he continues to organize research workshops and speak at scientific conferences around the world.

Thomas C. Gricks III

A prominent e-discovery lawyer and a leading authority on the use of technology-assisted review, Tom is the Director of Data Analytics at OpenText. At Catalyst he advised corporations and law firms on best practices for applying technology to reduce the time and cost of discovery and investigations. Tom has more than 25 years’ experience as a trial lawyer and in-house counsel, most recently with the law firm Schnader Harrison Segal & Lewis, where he was a partner and chair of the E-Discovery Practice Group. He was lead e-discovery counsel in Global Aerospace v. Landow Aviation, the first case in the country to authorize the use of TAR over the objection of opposing counsel.

Andrew Bye

Andrew is a machine learning solutions consultant at OpenText and a search and information retrieval expert. Throughout his career, Andrew has developed search practices for e-discovery and has worked closely with clients to implement effective workflows from data delivery through statistical validation. Before joining Catalyst, Andrew was a data scientist at Recommind. He has also worked as an independent data consultant and a linguist at H5. Andrew has a bachelor's degree in linguistics from the University of California, Berkeley and a master's in linguistics from UCLA.
# TABLE OF CONTENTS

Foreword by Andrew Peck .................................................. 6
Introduction to the Third Edition ......................................... 8

## A TAR Primer

1. Introduction Technology-Assisted Review .......................... 10
2. Types of TAR .................................................................. 17
3. How Do I Know if TAR Worked? ..................................... 26
4. How Much Can TAR 2.0 Save? ........................................ 34
5. Five Questions to Ask Your E-Discovery Vendor About CAL 42

## How Can I Use TAR

6. How Can I Use TAR 2.0 for More E-Discovery Tasks? ......... 48
7. Moving Beyond Outbound Productions: Implementing CAL for Knowledge Generation and Protection .................. 54
8. Using Contextual Diversity to Find Out What You Don't Know (and a Bit About Zipf's Law) ................................. 65
9. Using TAR for Multi-Language Discovery ......................... 75
10. What Are the Thresholds for Using Technology-Assisted Review? 81

## Putting TAR to Work

11. How Good Is That Keyword Search? Maybe Not as Good as You Think 86
12. TAR 2.0 vs. TAR 1.0: How Much Can You Save? ............... 95
13. Using Continuous Active Learning to Solve the Transparency Issue in TAR .................................................... 108
14. What Roles Do SMEs and Reviewers Play in a TAR 2.0 World? 115
15. An End to Family Batching: Reviewing on a Document Level Is More Efficient 120

## Validating a TAR Review

17. Measuring Recall for Discovery: An Introduction to Sampling 131
19. With CAL How Do I Know When to Stop Review? ............... 145
20. How Do I Validate My Results? ....................................... 151

## Case Studies

22. Major Bank Slashes Review Costs with Innovative E-Discovery Technology .............................................. 170
23. Predict Proves Effective for Small Collection .................... 175
25. Using Insight Predict for Review of Rolling Productions ... 184
26. 57 Ways to Leave Your (Linear) Lover ............................ 288
27. Predict Proves Effective Even With High Richness Collection 201
Foreword

By Andrew Peck

When I was a young lawyer, we did eyes on review. But we did so because we had no choice. There was no technology to help—unless one counts Post-it Notes that we stuck on documents. Unfortunately, as a result, lawyers of a certain age think eyes on review is the gold standard. It is not, as studies clearly show. In any event, with the volume of electronically stored information (ESI), eyes on review of everything is not feasible. It is too expensive and takes too long.

So lawyers began using keyword searching. While it is possible to do a good job in choosing keywords, unfortunately most lawyers do a poor job of choosing keywords. Fortunately there is a better way to review ESI: technology-assisted review, aka TAR. Scientific studies show that TAR is at least as efficient and effective as manual or keyword review at a fraction of the cost, and often is far superior to other methods.

But lawyers said we don't want to use TAR until there is judicial acceptance. I solved that concern in February 2012 with my decision in *Da Silva Moore*, the first case anywhere to accept use of TAR in appropriate cases. Since my Da Silva Moore decision, TAR has been accepted in other decisions in the United States, as well as Ireland, the United Kingdom and Australia.

Indeed, by 2015, reviewing the case law, I was able to say in Rio Tinto that TAR can no longer be considered an “unproven technology,” and that “the case law has developed to the point that it is now black letter law that where the producing party wants to utilize TAR for document review, courts will permit it.” I also noted in that case that the earlier concern that the opposing party would fight over the seed set was eliminated by the newer TAR method, continuous active learning, aka CAL.
So it baffles me that attorneys still seem to overwhelmingly use keywords instead of TAR. I don’t know why. Perhaps they don’t understand the technology. If so, this book is perfect in educating counsel and the clients to how TAR works, why it works and the benefits in terms of costs, time and efficiency.

We cannot be afraid of technology. I hope you will read this book and use TAR.

–Andrew Peck

The Honorable Andrew J. Peck served for 23 years as a United States Magistrate Judge for the Southern District of New York, including a term as Chief Magistrate Judge from 2004 to 2005. Before his appointment to the bench, Judge Peck was in private practice for 17 years, focusing on commercial and entertainment litigation, including copyright and trademark matters, with extensive trial experience. He now serves as Senior Counsel to DLA Piper.

Footnotes


Introduction

Third Edition

We published the first edition of *TAR for Smart People* in January 2015, just in time for Legaltech New York. As I wrote in the original introduction, the title was a play on the “For Dummies” series of books. In my experience in e-discovery, I haven't run into any dummies. So we addressed the book to all the smart people in the field.

In the years since, we have been surprised and encouraged by the positive reception the book has received. To be sure, we knew there were a lot of smart legal professionals out there looking to learn more about this important subject. What wasn’t clear to us was whether they were ready for a book like this, focusing on the latest TAR 2.0 protocols in a straightforward but technically sophisticated manner.

I am pleased to report that the response to the book surpassed our expectations, leading us to publish a second edition in 2016 and now a third in 2018. Indeed, the book has traveled farther and wider during its short life than I had imagined. We count thousands of readers from law firms, corporations and vendors across the United States, Asia and the European Union.

*TAR for Smart People* has been used as a text for a number of e-discovery courses helping law students learn about this important subject. Judges from both federal and state courts have used it at judicial conferences to help their colleagues get up to speed on the latest TAR methods. Practitioners from some of the largest and most prestigious law firms in the world have told me they keep it on their desks for reference.

More importantly, over the past few years, TAR 2.0 and its continuous active learning (CAL) protocol has moved from industry upstart to the
de facto standard for predictive analytics. The ball started rolling in 2014 when attorney Maura Grossman and Professor Gordon Cormack published landmark research demonstrating the superiority of CAL over earlier TAR 1.0 protocols. Momentum grew as other scientists showed similar results across hundreds of cases, with CAL beating one-time training every time.

Today, savvy TAR users understand the cost savings and time benefits of CAL over one-time training. They also appreciate the practical benefits of this newer protocol. These include the ability to handle rolling collections, to have reviewers rather than senior lawyers do training, and an uncanny ability to find relevant documents when collection richness is low.

I am proud to say that the third edition of *TAR for Smart People* represents a substantial rewrite and reorganization of our seminal work. A lot has happened in the past three years and we have learned a lot in that time, as has the industry. We offer this book as a guide to modern TAR written for both novices and experts alike.

In reorganizing this book, we wanted to not only expand its coverage but also to make it even more comprehensive for the smart people that we admire. Along with a new structure, you will find new chapters and case studies to supplement our earlier material, all in an effort to make this tricky subject even more comprehensible and enjoyable. Whether we achieved all these lofty goals is something for you to decide.

If you read previous editions of this book, we hope you will enjoy the content we have added along with a new take on old friends. If this is your first time reading the book, then we hope you enjoy this third edition even more. The subject is vital for all e-discovery professionals. There is simply no easier or more obvious way to cut review costs out there.

TAR for dummies? It turns out there were a lot more smart people out there after all.

*–John Tredennick, Esq.*

Founder, Catalyst
Technology-assisted review (TAR), aka predictive coding, predictive ranking or computer assisted review, is a process whereby humans interact with a computer to find relevant documents. Just as it has many names, there are many approaches to TAR. At bottom, however, all of these systems leverage human judgments about documents to find more relevant ones.

The process is iterative. A human reviews and tags a document as relevant or non-relevant. The computer, actually a computer algorithm, takes the human input and uses it to draw inferences about other as yet unseen documents. It then ranks them by likelihood of relevance and queues them up for the review team.

The process continues (review, analyze and rank) until your search objective is reached and the review finishes. Humans, rather than the computer, decide where to draw the cutoff, if any, and how many documents ultimately need to be reviewed. It all depends on your review objective.
The potential for savings in time and cost—without sacrificing quality—is what makes TAR revolutionary. Review teams can work faster using an ordered review because they are seeing the more important documents first, which often have similar content. Clients save on review costs because TAR provides a reasonable basis to stop review once most of the relevant documents have been found.

The savings in review time and costs can be substantial, which is why the topic is important. In some cases, a TAR process can cut out as much as 90% of the document population, which is a huge savings over linear review. You can defend the decision to cut off review through a relatively simple sampling process, which shows the percentage of relevant documents found, as well as the cost to find additional relevant documents in the unreviewed portion. Courts have accepted the reasonableness of this argument in dozens of cases, almost without exception.

**How Does It Work?**

A simple way to understand TAR is to think about Pandora Radio. Pandora has millions of songs in its archive but no idea what kind of music you like. Its goal is to play music from your favorite artists as well as new songs you haven’t heard but might like.
How does Pandora do this? You start by giving Pandora the name of a favorite artist, thus creating a station. Pandora will play a song from the artist you selected. Then, it might choose a similar song from another artist to see if you like it. You answer by clicking a “thumbs up” or “thumbs down” button. Information retrieval (IR) scientists call this relevance feedback.

Pandora analyzes the songs you like, as well as the songs you don’t, to make new suggestions. It looks at factors such as melody, harmony, rhythm, form, composition, style and vocalist to find similar songs. As you give it feedback on its suggestions, it uses that information to make better selections the next time. The IR people would call this training.

The process continues back and forth as you listen to your station. The more feedback you provide, the smarter the system gets. The end result is Pandora plays a lot of music you like and, occasionally, something you don’t like.

**Pandora for Documents?**

TAR operates much like Pandora, only you work with documents rather than songs. As you train the system, it gets smarter about which documents are relevant to your inquiry and which are not. It really is as simple as that.

Of course, TAR involves more serious objectives than choosing the music you want to hear, so there are options, techniques and strategies to consider. Also, vendors approach TAR in different ways, which can cause confusion for users. Not all TAR approaches are equally effective or simple to use but the brochures won’t tell you that.

Here is a start toward explaining how TAR works, written for people new to the process. We will get into more detail about the process and how it works later in the book.
1. Collect the documents you want to review.

To start, the computer has to analyze the documents you have collected for review, just like Pandora needs to analyze all the music it maintains. While approaches vary, most systems analyze the words in your documents in terms of the frequency in which they appear, both in each document and across the entire document population.

Some systems require that you collect all of the documents before you begin training. Others, particularly the newer TAR systems, allow you to add documents during the review process. This can be helpful in discovery because document collection often takes place over time.

2. Start training/review.

You have two choices here. You can start by presenting documents you know are relevant (or non-relevant) to the computer. Or you can let the computer select documents for consideration.

With Pandora, you begin by identifying an artist you like. This gives the computer a head start on your music preferences. In theory, you could let Pandora select music randomly until you found what you liked but this would be an inefficient way to get started.

With TAR, you go out and find starter documents using any means at your disposal (or even create a synthetic document). From these examples, the system begins to learn about your search objectives—analyzing which terms tend to occur in relevant documents and which occur in non-relevant ones. It then develops a model to predict the relevance of other documents in the population.

The documents you use to start the training process are called training seeds.

SME? There is an ongoing debate about whether training should be done by a senior lawyer, sometimes called a subject matter
expert (SME), to be effective. The answer may depend on the process you are using.

Early TAR 1.0 systems separated training from review, requiring that a SME do the initial training. Modern TAR 2.0 systems combine training and review in a single process which continues until the review is finished. As the review team tags documents, the algorithm keeps learning and reranking.

Our research suggests that review teams are generally as effective as SMEs in identifying relevant documents during a review. Waiting for a SME to review thousands of documents for training purposes delays the review and makes it more expensive than it needs to be.

3. Rank the documents by relevance.

This is the heart of the process. Based on the training you have provided, the system creates a model that it uses to rank your documents by relevance. Review then proceeds in ranked order, rather than by date, custodian or Bates number. Reviewing by relevance is what makes the TAR process so useful.

How does the algorithm do this? The answer will differ from one system to another and is based on complex math that goes beyond most lawyers’ ability to comprehend. The good news is that it doesn't matter how the algorithm works so long as we can objectively measure its effectiveness. We do that through sampling, which we will discuss in detail later in this book.

4. Continue training/review (rinse and repeat).

Continue training using your SME or review team. Most modern TAR systems will suggest additional documents for training, which helps the algorithm get better at understanding your document population. This is called active learning. Older TAR systems would select additional training documents randomly, which is called passive learning.

In general, the more training you do, the better the system will be at ranking unseen documents.
5. Test the ranking.

How how far have we gotten in the review? Are we halfway through or almost finished?

Depending on the type of TAR you are using, there are a number of ways to answer this question. One method is to take a sample of the unseen documents and use it to estimate how many relevant documents are left. You can compare that number with the number of relevant documents already seen to determine what percentage of relevant documents have been found.

With a TAR 2.0 system, you can compare the number of relevant documents in the daily review batches. When the percentage of relevant documents per batch drops significantly, it is a good sign you are nearing completion.

If you determine you have reviewed or identified a reasonable percentage of relevant documents, you may decide to cut off your review. If not, continue the review until your reach your objective. If you choose to review all of the documents, that is called a prioritized review.

6. Finish the review.

The end goal is to finish the review as efficiently and cost-effectively as possible. In the TAR context, “finishing” means reviewing until you have found “enough” relevant documents, with the concept of proportionality taking center stage.

Thus, you may have found enough documents after reviewing the first 20% of the ranking because you have seen 80% of the relevant documents. Your argument is that the cost to review the remaining 80% of the documents just to find the remaining 20% of the relevant documents is disproportionate and unduly burdensome. As we mentioned earlier, all reported court decisions accept this argument as reasonable and proportionate.

That's really all there is to it. While there are numerous options, techniques and strategies to be considered, it begins with these six basic steps.
Footnotes

1. IR specialists call these documents “relevant” but they do not mean relevant in a legal sense. They mean important to your inquiry even though you may not plan on introducing them at trial. You could substitute “hot,” “responsive,” “privileged” or some other criterion depending on the nature of your review.

2. We could use “irrelevant” but that has a different shade of meaning for the IR people so I bow to their use of non-relevant here. Either word works for this discussion.
As we mentioned in chapter one, there are a number of approaches to TAR based on different algorithms and varying workflows. Six years ago, in an attempt to simplify things, we described TAR systems as either TAR 1.0 or TAR 2.0.

Was this just marketing speak? Not at all. Despite individual differences, the first generation of TAR systems are based on one-time training. In contrast, TAR 2.0 systems are based on continuous learning, with the algorithm getting smarter as the review team progresses.

In this chapter, we will show you how these different TAR protocols work, drawing from the general TAR workflow we described in chapter one. We will also get into the specifics and talk about their strengths and weaknesses.

**TAR 1.0: One-Time Training**

The hallmark of TAR 1.0 is one-time training. In essence, a senior lawyer or subject expert tags a reference set and then reviews a few thousand documents to train the algorithm against that reference set. The trained algorithm then runs over the whole document population and the team reviews the portion that is deemed likely
relevant. The algorithm does no further training and cannot take advantage of the review team’s tags to get smarter about the unseen population.

Here are the typical steps in a TAR 1.0 process (with a little more detail than we used in chapter one):

1. **Collection**: Collect and process the documents for review. The TAR engine analyzes the text in the documents to determine similarities and differences between them.

   At this point the system knows nothing about your search goals. Rather, it is focused on grouping similar documents together in an initial ranking.

2. **Control set**: A subject matter expert (SME), usually a senior lawyer, reviews and tags a random sample (500+ documents) from the population to use as a control set for training.

   The control set functions as the “gold standard” for testing the ranking/classification algorithm, with the assumption that these documents have been correctly tagged as relevant or non-relevant. There is a second and somewhat questionable assumption here as well: that the text of these documents is representative of the larger document population.

3. **Initial seeds**: The SME begins the training process by submitting one or more tagged documents found through search or other means, or by reviewing and tagging a series of randomly selected documents.

   These are often called initial training seeds. The TAR engine uses
the text in these documents to start the training and ultimately to build a model that can be used to identify more relevant documents.

4. **Training:** The SME continues the training process by reviewing batches of documents selected randomly or chosen by the TAR engine. Each document is tagged as relevant or non-relevant.

The training rounds typically involve review of between 1,500 and 5,000 documents dished up to the SME in smaller batches. One leading TAR 1.0 system suggests as a starting point that the SME review about 40 batches of 40 documents each.

This training takes time. Using 60 documents an hour as a standard review rate, it may take the SME 65 hours to review and tag these 4,000 or so documents (a 500 document control set, 3,000 documents for training and another 500 for final testing).

5. **Ranking and testing:** At the end of each training round, the TAR algorithm analyzes the SME’s tags and continues building its relevance model. It tests the model by applying it to the documents in the control set to see how well it matched the SME’s judgments.

Thus, if 87 of the 500 documents in the control set are marked relevant, how close did the model come to identifying those 87 documents?

6. **Stability:** Training, ranking and testing continue until the algorithm’s model is “stable.” That means it is no longer getting better at identifying relevant documents in the control set.

As an example, say the model correctly identified 75 of the 87 relevant documents in the control set. Over a few more rounds of training, the results don’t improve. The algorithm may have reached its limit at finding relevant documents in the control set.

7. **Rank the remaining documents:** When training is complete, the next step is to run the model against the entire document population. Doing so can take several hours depending on the system, or it may need to run overnight.
This is a one-time ranking based on SME training. Once that work completes, the algorithm is not given any more documents for training and thus the algorithm can no longer improve based on further tagging by the review team.

8. **Test the ranking:** As a final quality control (QC) step, the SME reviews a 500 document sample of ranked—but not yet seen—documents to determine how well the algorithm did at finding relevant documents (or missing them). The purpose of the sample is to confirm that the algorithm’s ranking is reliable.

   At this point, the ranking can be used to determine a cutoff based on the desired level of recall. For example, say you determine that 80% recall (i.e. 80% of the relevant documents) is sufficient. You can estimate from the final ranking how many documents need to be reviewed to reach this goal.

9. **Conduct your review:** Once everything is done, the review team can be directed to look at documents with relevance scores higher than the cutoff point. Or you may decide to produce documents ranked above the cutoff without further review.

   You can also do a prioritized review in which the team looks at all of the documents collected, but does so based on their relevance ranking. That accomplishes two goals. First, if relevant documents are pushed to the top of the ranking, the team will see more important documents first. Second, once the team runs out of relevant documents, it can move quickly through the non-relevant ones without fear of missing something important.

**Differences in TAR 1.0 Protocols**

In a landmark study,¹ which we will discuss in later chapters, two leading TAR experts, Maura Grossman and Gordon Cormack, divided TAR 1.0 systems into two categories: simple passive learning (SPL) and simple active learning (SAL).

While they generally follow the above steps, the two TAR 1.0 approaches differ in the way documents are selected to train the systems.
**SPL** uses randomly-selected documents for training. The word passive refers to the fact that neither the algorithm nor the SME identifies documents for further training.

**SAL** relies on the algorithm to select the documents used for training. Typically the algorithm selects documents it is least sure about, in an attempt to identify the boundary between relevant and non-relevant documents.

Again, for both of these protocols, all training is done by the SME at the beginning of the process. Once the training is complete, the review size is fixed and cannot be reduced through further training.

**TAR 2.0: Continuous Active Learning**

Continuous active learning (CAL) is the hallmark of a TAR 2.0 protocol. A CAL system continually learns as the review progresses, and regularly reranks the document population based on what it has learned. As a result, the algorithm gets smarter and the team reaches its goal sooner, reviewing fewer documents than would otherwise be the case with one-time training.

Here is how the protocol works:

1. **Collection**: Collect and process the documents for review. The TAR algorithm analyzes the text in the documents to determine similarities and differences between them.

   Unlike in TAR 1.0 systems, you do not need to collect all of the
documents before beginning review. Simply add them to the system as they come in and they will be added to the ranking.

2. **Initial seeding:** Start by finding as many relevant documents as possible and feed them to the system to help train the algorithm. Or create a synthetic document to use as an initial seed.

   We don’t recommend using random sampling as a method to start training. It has been proven to be inefficient and is particularly problematic for low richness collections.

3. **Begin review:** The review team can start immediately. They will quickly begin seeing highly relevant documents. We also add a few documents to each batch selected by the computer based on contextual diversity. (We discuss later in this book.)

4. **QC:** As the review progresses, have a more senior attorney QC a small percentage of the documents. Our QC algorithm will present documents that are most likely tagged incorrectly.

5. **Finish:** Continue until you reach the desired recall rate. Track your progress as the review progresses to see when it is time to stop.

   You can demonstrate success through a systematic random sample of the unseen documents. It will show you where you are in the review and, where appropriate, how many more documents are needed to reach your goal.

The process is flexible. Start with as many training seeds as you like or create a synthetic document. After the initial ranking, the team can get going on the review. As they finish batches, the ranking engine takes their new judgments into account and keeps getting smarter. The end result is that you have fewer documents to review, and often a lot fewer, than you would with a TAR 1.0 system.

**Other TAR Protocols?**

In the years since we first coined the terms TAR 1.0 and TAR 2.0 as a simple method to distinguish the major TAR systems, a few writers have discussed what they call new protocols, calling them TAR 3.0 or
Predictive Coding 3.0, and later upgrading it to Predictive Coding 4.0. You can read about these systems on the web but from our study, they all fall under the TAR 2.0 ambit. All are based on continuous learning with simple differences such as adding clustering for training or integrating different search methods to find training seeds.

**TAR 1.0 Limits**

Regardless of the protocol used (SPL or SAL), the TAR 1.0 process comes with a number of practical problems that limits its effectiveness.

1. **One-time training:** A key problem with TAR 1.0 systems is that you get only “one bite at the apple.” Once the team starts their review, there is no opportunity for further training. A continuous learning process allows you to keep improving the algorithm. As a result, the team can review fewer documents to reach the same level of recall.

2. **Must collect everything:** To be statistically valid, you must collect all of the documents to be reviewed before training. If new documents surface, which is common in e-discovery, the initial training and control set become invalid.

3. **SMEs required:** A third problem is that TAR 1.0 requires a senior lawyer or subject matter expert (SME) to review thousands of marginally-relevant documents to build a control set and to train, test and validate the model. Not only is this expensive, but it delays the review until you can convince your busy senior attorney to sit still and get through the training.

4. **Training based on control set:** SME training is run against a (usually) 500-document control set. The statistically invalid assumption is that these 500 documents textually represent the much larger review population.

5. **Uses random or computer generated seeds to train the system:** This limits the use of key documents found by the trial team for training. Lawyers typically find relevant documents
quickly through search and client interviews. These often can't be used (or their use is severely limited) for training in TAR 1.0 systems.

6. **Won't work in low richness:** TAR 1.0 algorithms do not work well for low richness collections (e.g., a collection with only a few relevant documents). The problem with low richness is that it can be hard to find positive training examples for the control set or for training. If, for example, richness is below 1%, you may have to review several thousand documents just to find enough relevant ones to train the system. Indeed, this issue is sufficiently difficult that many TAR 1.0 vendors suggest their products shouldn't be used for low richness collections.

**TAR 2.0 Solutions**

TAR 2.0 and CAL address many of the problems associated with early TAR protocols. Here is a grid that shows how TAR 2.0 addresses the major TAR 1.0 weaknesses.

<table>
<thead>
<tr>
<th>TAR 1.0</th>
<th>TAR 2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. One-Time Training</strong> do not allow for training or learning past the initial training phase.</td>
<td><strong>1. Continuous Active Learning</strong> allows the algorithm to keep improving over the course of review, improving savings and speed.</td>
</tr>
<tr>
<td><strong>2. Must Collect Everything</strong> to be statistically valid. You must have all the documents to be reviewed before training. If new documents surface, which is common in e-discovery, the initial training and control set become invalid.</td>
<td><strong>2. Rolling Data Loads Are No Problem</strong> with continuous active learning. Newly added documents are immediately ranked and eligible for review, and a new random sample will provide updated metrics. With continuous training and no control set, metrics are not a problem after new data is added midstream.</td>
</tr>
<tr>
<td><strong>3. SMEs Required</strong> to handle all training. Review team judgments are not used to further train the system.</td>
<td><strong>3. Review Teams Train</strong> as they review, working alongside SME for maximum effectiveness. SME can focus on finding relevant documents and performing QC on review team judgments.</td>
</tr>
<tr>
<td><strong>4. Trains Against Small Control Set</strong>, limiting its ability to handle rolling uploads. Assumes all documents are received before ranking. Stability is based on comparison with reference set.</td>
<td><strong>4. Analyzes and Ranks Entire Collection Every Time</strong>, which allows rolling uploads. Does not use a reference set, but rather evaluates performance using multiple measures across the entire population.</td>
</tr>
</tbody>
</table>
These differences have led early TAR proponents to claim that the process is only suitable in cases with more than 100,000 documents. In sharp contrast, you can effectively use CAL in cases with only a few thousand documents, as we show later in the case studies. The team, or even a single reviewer, can simply start looking at documents (with or without initial seeds) and let the ranking move relevant documents to the top. The savings in time and costs by cutting a 3,000 document review to 1,000 documents is enough reason to standardize on a TAR 2.0 protocol.

Ultimately, TAR 2.0 beats TAR 1.0 on simplicity, low training overhead and cost savings. With TAR 1.0, an expert has to undertake a rigorous training process involving thousands of documents (control set, training and final QC) before review can begin. We will talk further about each of these points in the following chapters.

Footnotes


2. TAR 1.0 systems require a minimum number of relevant seeds for training to work.
How Do I Know if TAR Worked?

So far, we have talked about how technology-assisted review (TAR) works and about the different types. The next step is to determine whether and how well the process works.

How do we know a TAR project is successful? And how do we know when it is complete? Those are important questions that go to the heart of this book.

Was TAR Successful?

Answering this question is pretty easy. A TAR project is successful to the extent that the document ranking places more relevant documents at the front of the pack than you would get when the documents are ordered by other means (e.g. by date or Bates number). How successful you are depends on the degree to which the ranking is better than what you might get using your traditional approach.

Let us offer an example. Imagine your documents are represented by a series of cells, as in the below diagram. The shaded cells represent relevant documents and the white cells non-relevant.
What we have is a random distribution, or at least there is no discernable pattern between relevant and non-relevant. In that regard, this might be similar to a review in which you ordered documents by Bates number or date. In most cases, there is no reason to expect that relevant documents would appear at the front of the order.

With a random distribution such as the above, if you review 10% of the documents, you likely will find 10% of the relevant documents. If you review 50%, you will likely find 50% of the relevant documents.

Take a look at this next diagram. It represents the outcome of a perfect ordering. The relevant documents come first, followed by non-relevant documents.

If we could be confident that the ranking worked perfectly, as in this example, it is easy to see the benefit of ordering by rank. Rather than review all of the documents to find the relevant ones, you could simply review the first 20% and be done. You could ignore the remaining 80% (perhaps after sampling them) or, at least, direct them to a lower-priced review team.

**Yes, but What Is the Ranking Really Like?**

Since this is directed at smart people, we are sure you realize that computer rankings are never that good. At the same time, they are rarely (if ever) as bad as you might see in a linear review.

Following our earlier examples, here is how the actual ranking might look using predictive ranking:

We see that the algorithm did a better job than a random distribution, although it is far from perfect. We have 30% of the relevant documents at the top of the order, followed by an increasing mix of non-relevant documents. At about a third of the way into the review, you would start to run out of relevant documents.

This would be a success by almost any measure. If you stopped your review at the midway point, you would have seen all but one relevant
document. By cutting out half the document population, you would save substantially on review costs.

**How Do I Measure Savings?**

If the goal of TAR is to arrange a set of documents in order of likely relevance to a particular issue, the measure of success is the extent to which that goal is met. Put as a question: Am I getting more relevant documents at the start of my review than I would with my typical approach (often a linear review). If the answer is yes, then how much better is it?

To answer these questions, we need to take two additional steps. First, for comparison purposes, we will want to measure the “richness” of the overall document population. Second, we need to determine how many relevant documents we have found at any point in the review.

1. **Estimating richness:** Richness is a measure of how many relevant documents are in your total document population. IR scientists call this “prevalence,” as a reference to how prevalent relevant documents are in the total population. For example, say we estimate that 15% of the documents in the population are relevant, with 85% non-relevant. We would say document richness, or prevalence, is 15%.

   How do we estimate richness? Once the documents are assembled, we can use random sampling for this purpose. In general, a random sample allows us to look at a small subset of the document population and make predictions about the nature of the larger set.

   Thus, from the example above, if our sample found 15 documents out of 100 to be relevant, we would project a richness of 15%. Extrapolating that to the larger population (100,000 for example), we might estimate that there were about 15,000 relevant documents to be found.

   For those really smart people who understand statistics, we are skipping a discussion about confidence intervals (aka margins...
of error). Let us just say that the larger the sample size, the more confident you can be in your estimate. But, surprisingly, the sample size does not have to be that large to provide a high degree of confidence. You can read more about the topic later in this book.

2. **Evaluating the ranking:** Once the documents are ranked, we can sample the ranking to determine how well our algorithm did at pushing relevant documents to the top of the stack. At Catalyst, we do this through a systematic random sample. Many other vendors use a simple random sample instead.

   In a systematic random sample, we choose documents based on their ranked order. Specifically, we sample every nth document from the top to the bottom of the ranking (e.g. every 100th document). Using this method helps ensure that we are looking at documents across the ranking spectrum, from highest to lowest. (Let us note that we do not present them in ranked order to ensure that the review is unbiased.)

   As an aside, you can use a systematic random sample to determine overall richness/prevalence as well as to evaluate the ranking. Unless you need an initial richness estimate for review planning purposes before any documents have been reviewed, we recommend you do both steps at the same time.
Comparing the Results

We can compare the results of the systematic random sample to the richness of our population by plotting what scientists call a “yield curve.” While this may sound daunting, it is really rather simple. It is the one diagram you should know about if you are going to use TAR.

A yield curve can be used to show the progress of a review and the results it yields, at least in the number of relevant documents found. The y-axis shows the percentage of documents reviewed or to be reviewed. The y-axis shows the percentage of relevant documents found (or you would expect to find) at any given point in the review.

**Linear review:** We can plot the likely outcome of a linear review by drawing a diagonal line across the graph going from zero to 100%. It reflects the fact that, in a linear review, we expect the percentage of relevant documents to correlate with the percentage of total documents reviewed.

Following that notion, we can estimate that if the team were to review 10% of the document population, they would likely see 10% of the relevant documents. If they were to look at 50% of the documents, we would expect them to find 50% of the relevant documents, give or take. If they wanted to find 80% of the relevant documents, they would have to look at 80% of the entire population.

This becomes the baseline for measuring the success of a TAR review.
**Predictive review:** Now let’s plot the results of our systematic random sample. The purpose is to show how the review might progress (or did progress) if we reviewed documents in a ranked order, from likely relevant to likely non-relevant. We can easily compare it to a linear review to measure the success of the predictive ranking process.

![Yield Curve](image)

You can see that the line for the predictive review goes up more steeply than the one for linear review. This reflects the fact that in a predictive review the team starts with the most likely relevant documents. The line continues to rise until you hit the 80% relevant mark, which happens after a review of about 10–12% of the entire document population. The slope then flattens, particularly as you cross the 90% relevant line. That reflects the fact that you won’t find as many relevant documents from that point onward. Put another way, you will have to look through a lot more documents before you find your next relevant one at this point in the ranking.

We have what we need to measure the success of our predictive ranking project. To recap, we needed:

1. A richness estimate so we have an idea of how many relevant documents are in the population.

2. A systematic random sample so we can estimate how many relevant documents got pushed to the front of the ordering.
It is now relatively easy to quantify success. As the yield curve illustrates, if I engage in a predictive review, I will find about 80% of the relevant documents after only reviewing about 12% of total documents. If I wanted to review 90% of the relevant documents, I could stop after reviewing just over 20% of the population. My measure of success would be the savings achieved over the linear review.

That equation is quite simple. If I stopped after reviewing 20,000 documents and did not review the additional 80,000 documents because they were beyond the cutoff, I saved on the review of 80,000 documents. If it cost me $1 per reviewed document, I just saved $80,000.

**About Precision and Recall**

There are two terms we should discuss in conjunction with our review of yield curves: precision and recall. They are both helpful to the discussion about TAR success and also come into play in the next section when we talk about stopping the review. Fortunately, both are simple, albeit important, terms.

**Precision:** Precision is the measure of how many relevant documents are found in a search or in the review set. For example, if your search brings back 500 documents but only 50 are relevant, we would say precision is 10%. Put another way, you have to look at 10 documents to find one relevant document.

In a review set, precision stands for the same thing. A review with 50% precision means the reviewer looks at two documents for each relevant one found. That is much more efficient than one with 10% precision. Without question, greater precision means you will find relevant documents more quickly and save on review costs.

**Recall:** Recall is a measure of the percentage of relevant documents found in a search or in a review process. If a search brings back 500 relevant documents out of 5,000 that exist in the population, we would say recall is 10%. Likewise, if the review team finds 750 relevant documents out of a total of 1,000 in the population, we would say recall is 75%.

Recall is one of the key measures of success in a discovery review.
While there are no hard and fast rules, and we don’t mean this to be a legal treatise, know that courts have approved discovery productions where recall was between 70–80%. Is that a sufficient result for your matter? It depends.

At this point we move into proportionality arguments. What is the right stopping point in a review? The answer depends on the needs of your case, the nature of the documents and any stipulations among the parties. At the least, being able to take a sample and understanding yield curves will help you frame the argument in a meaningful way.

---

Footnotes

1. We will use a linear review as a baseline because that is the way most reviews are done. In a linear review, the documents are ordered by criteria such as date or Bates numbers. That is no reason to believe that such an ordering will place relevant documents at the front of the review any more than a random ordering might. And, most linear reviews involve 100% of the document population.

2. Note that an estimate based on a random sample is not valid unless you are sampling against the entire population. If you get new documents, you have to redo your sample.
How Much Can TAR 2.0 Save?

A Look at the Grossman/Cormack Research Results

In an earlier chapter, we discussed different types of technology-assisted review (TAR), relying on an important study by Maura Grossman and Gordon Cormack that classified the leading TAR 1.0 protocols as either simple passive learning (SPL) or simple active learning (SAL). They also gave name to continuous active learning (CAL), a hallmark of TAR 2.0.

Perhaps the most important conclusion of the study was that in every case CAL proved to be more effective than SPL or SAL. To quote Grossman and Cormack:

“The results show that entirely non-random training methods, in which the initial training documents are selected using a simple keyword search, and subsequent training documents are selected by active learning [CAL], require substantially and significantly less human review effort . . . to achieve any given
level of recall, than passive learning, in which the machine-learning algorithm plays no role in the selection of training documents [SPL]. ... 

Among active-learning methods, continuous active learning with relevance feedback yields generally superior results to simple active learning with uncertainty sampling [SAL], while avoiding the vexing issue of “stabilization”—determining when training is adequate, and therefore may stop.”

How much can you expect to save using CAL over the simple passive and active learning methods used by TAR 1.0 programs? While every case is different, as are the algorithms that different vendors employ, we can draw some interesting conclusions from the Grossman/Cormack study that will help answer this question.

**Comparing CAL with SPL and SAL**

Grossman and Cormack compared the three TAR protocols against eight different matters. Four were from an earlier Text REtrieval Conference (TREC) program and four were from actually litigated cases.

After charting the results from each matter, they offered summary information about their results. In this case we will show them for a typical TAR 1.0 project with 2,000 training seeds.

<table>
<thead>
<tr>
<th>Matter</th>
<th>Collection Size</th>
<th>CAL</th>
<th>SPL</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>723,537</td>
<td>6,000</td>
<td>284,000</td>
<td>237,000</td>
</tr>
<tr>
<td>202</td>
<td>723,537</td>
<td>11,000</td>
<td>47,000</td>
<td>34,000</td>
</tr>
<tr>
<td>203</td>
<td>723,537</td>
<td>6,000</td>
<td>521,000</td>
<td>43,000</td>
</tr>
<tr>
<td>207</td>
<td>723,537</td>
<td>11,000</td>
<td>103,000</td>
<td>55,000</td>
</tr>
<tr>
<td>A</td>
<td>1,118,116</td>
<td>11,000</td>
<td>502,000</td>
<td>210,000</td>
</tr>
<tr>
<td>B</td>
<td>409,277</td>
<td>8,000</td>
<td>142,000</td>
<td>119,000</td>
</tr>
<tr>
<td>C</td>
<td>293,549</td>
<td>4,000</td>
<td>9,000</td>
<td>5,000</td>
</tr>
<tr>
<td>D</td>
<td>405,796</td>
<td>18,000</td>
<td>55,000</td>
<td>60,000</td>
</tr>
</tbody>
</table>

A quick visual inspection confirms that the CAL protocol requires the review of far fewer documents than required for simple passive or simple active learning. In Matter 201, for example, a CAL review requires inspection of 6,000 documents in order to find 75% of the relevant files. In sharp contrast, reviewers using a SPL protocol would have to view 284,000 documents. For SAL, they would have to review
almost as many, 237,000 documents. Both TAR 1.0 protocols require review of more than 230,000 documents. At $4 per document for review and QC, the extra cost from using the TAR 1.0 protocols would come to almost $1 million.

Some of the other matters had numbers that were much closer. Matter C, for example, required the review of 4,000 documents for a CAL protocol but only 5,000 for SAL and 9,000 for SPL. In such a case, the savings are much smaller, hardly justifying a switch in TAR applications. So what might we expect as a general rule if we were considering different approaches to TAR?

**Averaging the Results Across Matters**

Lacking more comparative data, one way to answer this question is to use the averages across all eight matters to make our analysis.

<table>
<thead>
<tr>
<th>Matter</th>
<th>Collection Size</th>
<th>CAL</th>
<th>SPL</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>723,537</td>
<td>6,000</td>
<td>284,000</td>
<td>237,000</td>
</tr>
<tr>
<td>202</td>
<td>723,537</td>
<td>11,000</td>
<td>47,000</td>
<td>34,000</td>
</tr>
<tr>
<td>203</td>
<td>723,537</td>
<td>6,000</td>
<td>521,000</td>
<td>43,000</td>
</tr>
<tr>
<td>207</td>
<td>723,537</td>
<td>11,000</td>
<td>103,000</td>
<td>55,000</td>
</tr>
<tr>
<td>A</td>
<td>1,118,116</td>
<td>11,000</td>
<td>502,000</td>
<td>210,000</td>
</tr>
<tr>
<td>B</td>
<td>409,277</td>
<td>8,000</td>
<td>142,000</td>
<td>119,000</td>
</tr>
<tr>
<td>C</td>
<td>293,549</td>
<td>4,000</td>
<td>9,000</td>
<td>5,000</td>
</tr>
<tr>
<td>D</td>
<td>405,796</td>
<td>18,000</td>
<td>55,000</td>
<td>60,000</td>
</tr>
<tr>
<td>Average</td>
<td>640,111</td>
<td>9,375</td>
<td>207,875</td>
<td>95,375</td>
</tr>
</tbody>
</table>

Our average matter size is just over 640,000. The CAL protocol would require review of 9,375 documents. With SPL you would have to review 207,875 documents. With SAL, you would have to review 95,375 documents. Clearly SAL is to be preferred to SPL but it still required the review of an extra 86,000 documents when compared to CAL.

How much would that cost? To determine this there are several factors to consider. First, the TAR 1.0 protocols require that a subject matter expert (SME) do the initial training. CAL does not require this. Thus, we have to determine the hourly rate of the SME. We then have to determine how many documents an hour the expert (and later the reviewers) can get through. Lastly, we need an estimate for reviewer costs.
Here are some working assumptions:

2. Cost for a standard reviewer: $60/hour.
3. Documents per hour reviewed (for both SME and reviewer): 60.

If we use these assumptions and work against our matter averages, we can compare the costs of using the three protocols. On an average review, at least based on these eight matters, you can expect to spend over a quarter million dollars in review costs if you use SPL as your TAR protocol. And if you use CAL, you can expect to save $115,000 over SAL. These are significant sums.

<table>
<thead>
<tr>
<th>Comparing CAL to SPL and SAL with 2,000 Training Seeds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matter</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>201</td>
</tr>
<tr>
<td>202</td>
</tr>
<tr>
<td>203</td>
</tr>
<tr>
<td>207</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>D</td>
</tr>
<tr>
<td>Average</td>
</tr>
<tr>
<td>Reviewed by SME</td>
</tr>
<tr>
<td>Expert review cost</td>
</tr>
<tr>
<td>Reviewer cost</td>
</tr>
<tr>
<td>Total review cost</td>
</tr>
</tbody>
</table>

What About Using More Training Seeds?

Grossman and Cormack also reported the results for the TAR 1.0 protocols when substantially more training seeds were used: 5,000 and 8,000. If your SME is willing to review this many training documents, you will get better results from TAR 1.0. That seems logical since the more training seeds used, the better the results should be.

The problem is that at 60 documents an hour, your SME will spend 83 hours (about two weeks) doing the training with 5,000 seeds. He/she would spend more than 133 hours (about 3.5 weeks) if you go for 8,000 seeds. Even worse, he/she may have to redo the training if new documents come in later.²
That said, here is how the numbers worked out for 5,000 training seeds:

<table>
<thead>
<tr>
<th>Matter</th>
<th>Collection Size</th>
<th>CAL</th>
<th>SPL (5,000)</th>
<th>SAL (5,000)</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>723,537</td>
<td>6,000</td>
<td>331,000</td>
<td>7,000</td>
</tr>
<tr>
<td>202</td>
<td>723,537</td>
<td>11,000</td>
<td>29,000</td>
<td>12,000</td>
</tr>
<tr>
<td>203</td>
<td>723,537</td>
<td>6,000</td>
<td>331,000</td>
<td>7,000</td>
</tr>
<tr>
<td>207</td>
<td>723,537</td>
<td>11,000</td>
<td>50,000</td>
<td>23,000</td>
</tr>
<tr>
<td>A</td>
<td>1,118,116</td>
<td>11,000</td>
<td>326,000</td>
<td>42,000</td>
</tr>
<tr>
<td>B</td>
<td>409,277</td>
<td>8,000</td>
<td>41,000</td>
<td>10,000</td>
</tr>
<tr>
<td>C</td>
<td>293,549</td>
<td>4,000</td>
<td>8,000</td>
<td>8,000</td>
</tr>
<tr>
<td>D</td>
<td>405,796</td>
<td>18,000</td>
<td>38,000</td>
<td>54,000</td>
</tr>
</tbody>
</table>

Average | 640,111 | 9,375 | 144,250 | 20,375 |
Reviewed by SME | 5,000 | 5,000 |

Expert review cost | $29,167 | $29,167 |
Reviewer cost | $11,719 | $174,063 | $19,219 |
Total review cost | $11,719 | $203,229 | $48,385 |
Savings from CAL | $191,510 | $36,667 |

And for 8,000 training seeds:

<table>
<thead>
<tr>
<th>Matter</th>
<th>Collection Size</th>
<th>CAL</th>
<th>SPL (8,000)</th>
<th>SAL (8,000)</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>723,537</td>
<td>6,000</td>
<td>164,000</td>
<td>10,000</td>
</tr>
<tr>
<td>202</td>
<td>723,537</td>
<td>11,000</td>
<td>26,000</td>
<td>14,000</td>
</tr>
<tr>
<td>203</td>
<td>723,537</td>
<td>6,000</td>
<td>154,000</td>
<td>10,000</td>
</tr>
<tr>
<td>207</td>
<td>723,537</td>
<td>11,000</td>
<td>36,000</td>
<td>13,000</td>
</tr>
<tr>
<td>A</td>
<td>1,118,116</td>
<td>11,000</td>
<td>204,000</td>
<td>12,000</td>
</tr>
<tr>
<td>B</td>
<td>409,277</td>
<td>8,000</td>
<td>21,000</td>
<td>11,000</td>
</tr>
<tr>
<td>C</td>
<td>293,549</td>
<td>4,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>D</td>
<td>405,796</td>
<td>18,000</td>
<td>37,000</td>
<td>53,000</td>
</tr>
</tbody>
</table>

Average | 640,111 | 9,375 | 81,500 | 16,625 |
Reviewed by SME | 8,000 | 8,000 |

Expert review cost | $46,667 | $46,667 |
Reviewer cost | $11,719 | $91,875 | $10,781 |
Total review cost | $11,719 | $138,542 | $57,448 |
Savings from CAL | $126,823 | $45,729 |

The first thing to notice is that the number of documents that ultimately have to be reviewed reduces as you add more training seeds. However, also note that the total review cost for SAL increases as you go from 5,000 to 8,000 training seeds. This is because we assume you have to pay more for SME training than review team training. With CAL, the reviewers do the training.

**How Much Time Can I Save?**

So far, we have only looked at cost savings. What about time savings? We can quickly see how much time the CAL protocol saves as well.
For 2,000 training seeds:

<table>
<thead>
<tr>
<th>Matter</th>
<th>Collection Size</th>
<th>CAL</th>
<th>SPL</th>
<th>SAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>723,537</td>
<td>6,000</td>
<td>284,000</td>
<td>237,000</td>
</tr>
<tr>
<td>202</td>
<td>723,537</td>
<td>11,000</td>
<td>47,000</td>
<td>34,000</td>
</tr>
<tr>
<td>203</td>
<td>723,537</td>
<td>6,000</td>
<td>521,000</td>
<td>43,000</td>
</tr>
<tr>
<td>207</td>
<td>723,537</td>
<td>11,000</td>
<td>103,000</td>
<td>55,000</td>
</tr>
<tr>
<td>A</td>
<td>1,118,116</td>
<td>11,000</td>
<td>502,000</td>
<td>210,000</td>
</tr>
<tr>
<td>B</td>
<td>409,277</td>
<td>8,000</td>
<td>142,000</td>
<td>119,000</td>
</tr>
<tr>
<td>C</td>
<td>293,549</td>
<td>4,000</td>
<td>9,000</td>
<td>5,000</td>
</tr>
<tr>
<td>D</td>
<td>405,796</td>
<td>18,000</td>
<td>55,000</td>
<td>60,000</td>
</tr>
</tbody>
</table>

Average: 640,111, 9,375, 207,875, 95,375
Reviewed by SME: 2,000
Review time (hours): 156
Time savings (hours): 3,308

For 5,000 training seeds:

<table>
<thead>
<tr>
<th>Matter</th>
<th>Collection Size</th>
<th>CAL</th>
<th>SPL (8,000)</th>
<th>SAL (8,000)</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>723,537</td>
<td>6,000</td>
<td>164,000</td>
<td>10,000</td>
</tr>
<tr>
<td>202</td>
<td>723,537</td>
<td>11,000</td>
<td>26,000</td>
<td>14,000</td>
</tr>
<tr>
<td>203</td>
<td>723,537</td>
<td>6,000</td>
<td>154,000</td>
<td>10,000</td>
</tr>
<tr>
<td>207</td>
<td>723,537</td>
<td>11,000</td>
<td>36,000</td>
<td>13,000</td>
</tr>
<tr>
<td>A</td>
<td>1,118,116</td>
<td>11,000</td>
<td>204,000</td>
<td>12,000</td>
</tr>
<tr>
<td>B</td>
<td>409,277</td>
<td>8,000</td>
<td>21,000</td>
<td>11,000</td>
</tr>
<tr>
<td>C</td>
<td>293,549</td>
<td>4,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>D</td>
<td>405,796</td>
<td>18,000</td>
<td>37,000</td>
<td>53,000</td>
</tr>
</tbody>
</table>

Average: 640,111, 9,375, 81,500, 16,625
Reviewed by SME: 8,000
Review time (hours): 156
Time savings (hours): 1,202

And, for 8,000 training seeds:

<table>
<thead>
<tr>
<th>Matter</th>
<th>Collection Size</th>
<th>CAL</th>
<th>SPL (5,000)</th>
<th>SAL (5,000)</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>723,537</td>
<td>6,000</td>
<td>331,000</td>
<td>7,000</td>
</tr>
<tr>
<td>202</td>
<td>723,537</td>
<td>11,000</td>
<td>29,000</td>
<td>12,000</td>
</tr>
<tr>
<td>203</td>
<td>723,537</td>
<td>6,000</td>
<td>331,000</td>
<td>7,000</td>
</tr>
<tr>
<td>207</td>
<td>723,537</td>
<td>11,000</td>
<td>50,000</td>
<td>23,000</td>
</tr>
<tr>
<td>A</td>
<td>1,118,116</td>
<td>11,000</td>
<td>326,000</td>
<td>42,000</td>
</tr>
<tr>
<td>B</td>
<td>409,277</td>
<td>8,000</td>
<td>41,000</td>
<td>10,000</td>
</tr>
<tr>
<td>C</td>
<td>293,549</td>
<td>4,000</td>
<td>8,000</td>
<td>8,000</td>
</tr>
<tr>
<td>D</td>
<td>405,796</td>
<td>18,000</td>
<td>38,000</td>
<td>54,000</td>
</tr>
</tbody>
</table>

Average: 640,111, 9,375, 144,250, 20,375
Reviewed by SME: 5,000
Review time (hours): 156
Time savings (hours): 2,248
As with cost savings, there are substantial review time savings to be had using CAL over simple passive learning and simple active learning. The savings range from 121 hours (SAL at 8,000 training seeds) to as much as 3,308 hours (SPL at 2,000 training seeds).

**So How Much Can I Save with CAL?**

“A lot” is the answer, based on the Grossman/Cormack research. We have published similar studies with similar results. Given this evidence, it is hard to imagine why anyone would use these out-of-date TAR protocols.

There are a number of other benefits that go beyond cost and time savings. First, CAL works well with low richness collections. While some populations have high percentages of relevant documents, not all do. Why not choose one protocol that covers both ends of the spectrum equally well?

Second, as mentioned earlier, the CAL protocol allows for the continuous addition of documents without need for costly and time-consuming retraining. Simply add the new documents to the collection and keep reviewing.

This is particularly true if you use our contextual diversity engine to find documents that are different from those you have already seen. Contextual diversity protects against the possibility of bias stemming from using documents found through keyword searches.

Third, review can begin right away. With TAR 1.0 protocols, the review team can’t begin until a SME does the training. Depending on the SME’s schedule and inclination to look at other documents, the review can be held up for days or weeks. With CAL, the review starts right away.

These are just a few ways in which TAR 2.0 solves real world problems. Why pay more in review costs and time to use an inferior protocol?
Footnotes


2. In a TAR 1.0 process, training runs against a control set, about 500 documents chosen at random from the larger population. If new documents are added to the population, the randomly selected control set is no longer valid. The same is true for the training effort and thus you may have to redo training.
In the aftermath of studies showing that continuous active learning (CAL) is more effective than the first-generation technology-assisted review (TAR 1.0) protocols, it seems like every e-discovery vendor is jumping on the bandwagon. At the least it feels like every e-discovery vendor claims to use CAL or somehow incorporate it into its TAR protocols.

Despite these claims, there remains a wide chasm between the TAR protocols available on the market today. As a TAR consumer, how can you determine whether a vendor that claims to use CAL actually does? Here are five basic questions you can ask your vendor to ensure that your review effectively employs CAL.

1. Does Your TAR Tool Use a Control Set for Training?

Control sets are the hallmark of TAR 1.0, but wholly inconsistent with the concept of CAL. In fact, the use of a control set for training can often impair and complicate the TAR process.
To train a TAR 1.0 tool, you typically start by generating a random sample to represent the entire document population to some statistical degree of certainty. That random sample—a small fraction of the entire collection—is considered the control set. As documents are reviewed and coded, the progress of training is measured against the control set, which is re-ranked after every training round. Once it appears that training is having little impact on the ranking of the control set, the tool is considered to be stabilized. The one-time training effort concludes and the tool is used to rank the entire collection for production review.

By comparison, a true CAL process uses no control set. Every review decision is used to train the tool, and the entire collection (not a small subset) is constantly re-ranked and monitored. Only when it appears that you have reached your goal vis-a-vis the entire collection, or that training is having no further impact on the ranking of the entire collection and responsive documents are no longer being returned for review, do review and training cease.

2. Does Training Focus on Marginally Relevant Documents or Highly Relevant Documents?

Training that focuses on marginally relevant documents will not optimize the use of CAL. In fact, TAR protocols that focus on marginally relevant documents are typically not CAL and are generally less effective.

The predominant objective of reviewing marginally relevant documents is to determine where best to draw the line between relevant and non-relevant documents. The ultimate goal is to train an algorithm, called a “classifier,” to make that distinction, so the presumptively relevant documents can be separately reviewed for production. Generally, TAR protocols that use a classifier to segregate documents neither rank the collection nor train continuously through the attainment of review objectives. Thus, they would not be considered CAL.

As illustrated by the below chart, the classifier approach has two drawbacks. First, no matter how well the line is drawn, some number of relevant documents (either cats or dogs) will fall on the other side of the line and never be seen in the review set. Second, the tighter
you try to draw that line, the more time and effort it takes before you can even begin to review documents.

Conversely, the objective of CAL is to continuously use reviewer judgments to improve training and rank the collection, and to use the improved ranking to present the reviewers with better documents. This process continues iteratively until the review targets are achieved for the collection as a whole. To implement this protocol effectively, training and review focus primarily and specifically on highly relevant documents. There is no wasted effort, and every relevant document is available for review.

3. How Often Do You Rank the Collection During Review?

The essence of CAL is its ability to harness reviewer judgments to rank the collection and return the best documents to the reviewers as early as possible. Every time CAL ranks the collection, reviewer judgments are leveraged to improve review and, in turn, to improve the next ranking. The process is cyclical and the results exponential.

Studies prove the obvious—the more frequent the ranking, the better the results. This phenomenon is akin to compounding interest. The more frequently interest is compounded, the more rapidly the benefit accrues. With CAL, the more frequently the collection is ranked, the more rapidly reviewers can take advantage of their
collective decisions to successively feed judgments back to the tool for further refinement.

Among vendors, a tremendous disparity exists in the frequency with which they rank the collection (or the control set, with TAR 1.0). Catalyst, for example, can and does rank millions of documents in minutes. Reviewer judgments are used to rank the entire collection several times every hour throughout the review and training process. Most other vendors rank only the control set (not the collection) during review and training, and subsequently rank the entire collection only once. Even worse, their process of ranking the entire collection can typically take several hours to complete.

4. Is It Necessary to Have a Subject Matter Expert Do the Training?

Although TAR 1.0 requires a subject matter expert (SME) for training, CAL does not. In fact, since all review is training with CAL, training by an SME would be prohibitive. CAL frees the SME to focus on more productive tasks and leave the bulk of training to the reviewers. This enables immediate review and eliminates the time and expense associated with training a TAR 1.0 tool.

With TAR 1.0, training is a one-time effort, the results are driven by comparison to a finite control set, and the process dictates exactly which documents will be reviewed for production. This creates an inherent need for the consistency of a single decision maker with the knowledge and authority to establish the scope of the eventual review.

This is not the case with CAL, where every review judgment trains the tool. Reviewers see the same documents they would have seen after training using a TAR 1.0 protocol (perhaps more), and presumably make the same decisions. Because the tool is continuously learning from the reviewers’ judgments, the universe of documents passed to the reviewers is constantly refined to elevate those most likely to be produced.

The upshot of eliminating SME review is savings—of both time and money. TAR 1.0 typically requires at least two weeks of SME effort before the review team can review a single document. Given the
billing rate of senior attorneys most likely to serve as SMEs, that effort will cost tens of thousands of dollars. With CAL, review and training are coextensive and start immediately, with no sunk cost for training.

5. What Is Your Process for Handling Rolling Collections?

The reality of modern discovery is that all of the documents to be reviewed for production are rarely available at the same time. Instead, documents are collected piecemeal and arrive on a rolling basis. An added benefit of CAL is its ability to incorporate new documents into the collection at virtually any point in the review without sacrificing previous effort. If the vendor suggests that a rolling collection presents any impediment to seamless review, the vendor is not making an efficient or effective use of CAL.

Rolling collections are a problem for TAR 1.0 protocols because they rely on control sets for training. A control set is intended to represent the entire collection. Since newly added documents change the character of the collection, the initial control set is no longer representative. Every time documents are added to a collection, a new, revised or additional control set needs to be generated. Even worse, if new documents are added after training is completed and a review set generated, it may be necessary to completely retrain the tool in addition to preparing a new control set.

CAL is not subject to these limitations. As new documents are received, they are simply incorporated into the collection and integrated according to the current ranking. As those documents are reviewed and coded through the continuous learning process, the ranking adjusts to reflect the new information. No effort is lost. Every previous judgment remains intact and every subsequent judgment further improves the ranking.

Bonus Question: How Easy Is It to Run Simultaneous TAR Projects?

Another benefit of CAL is the ability to run simultaneous TAR projects and generate useful results almost immediately, at any point during
the review and with little to no additional setup. With TAR 1.0, the process is much more cumbersome. If your vendor does not allow you to easily and quickly implement simultaneous TAR projects, you are not using CAL to its fullest potential.

Since review is training with CAL, very little is required to run simultaneous TAR projects covering different issues. Simply identify each of the pertinent issues and code the documents for each issue during review. The tool will use each judgment, and generate and maintain a separate ranking for each issue. Once you attain the review objective for one TAR project, you can focus on the next project. The existing ranking will make every successive project more efficient.

CAL can incorporate new TAR projects at any point during review and quickly generate results. Review and coding for a new project can start as soon as a new issue is identified. From that point forward, every review decision can include a judgment on the new issue. By focusing specifically on the new TAR project, review and training will quickly improve the ranking and return the best documents for further review.

TAR 1.0 is too cumbersome to do this effectively. With TAR 1.0, every project requires a separately coded control set against which to evaluate training. This makes simultaneous projects, especially new projects arising during review, difficult to implement.

CAL provides significant advantages over other TAR protocols, in both efficiency and effectiveness. So how can you be sure that your vendor is actually equipping your review project with all of the benefits of an efficient and effective CAL protocol? Just ask five simple questions—and throw in the bonus for good measure.
How Can I Use TAR 2.0 for More E-Discovery Tasks?

Recent advances in technology-assisted review (TAR 2.0) include the ability to deal with low richness, rolling collections, and flexible inputs in addition to vast improvements in speed. These improvements now allow TAR to be used effectively in many more discovery workflows than traditional TAR 1.0 systems, which were primarily used to classify large numbers of documents for production.

To better understand this, it helps to begin by examining in more detail the kinds of tasks lawyers face. Broadly speaking, document review tasks fall into three categories:

- **Classification.** This is the most common form of document review, in which documents are sorted into buckets such as responsive or non-responsive so that we can do something different with each class of document. The most common example here is a review for production.

- **Protection.** This is a higher level of scrutiny in which the purpose is to protect certain types of information from disclosure. The most common example is privilege review, but this also encompasses trade secrets and other forms of confidential,
protected or even embarrassing information, such as personally identifiable information (PII) or confidential supervisory information (CSI).

- **Knowledge Generation.** The goal here is learning what stories the documents can tell us and discovering information that could prove useful to our case. A common example of this is searching and reviewing documents received in a production from an opposing party or searching a collection for documents related to specific issues or deposition witnesses.

You’re probably already quite familiar with these types of tasks, but we want to get explicit and discuss them in detail because each of the three has distinctly different recall and precision targets. In turn, they have important implications for designing your workflows and integrating TAR.

**Metrics**

Let’s quickly review those two crucial metrics for measuring the effectiveness and defensibility of a discovery process, “recall” and “precision.” Recall is a measure of completeness, the percentage of relevant documents actually retrieved. Precision measures purity, the percentage of retrieved documents that are relevant.

The higher the percentage of each, the better you’ve done. If you achieve 100% recall, then you have retrieved all the relevant documents. If all the documents you retrieve are relevant and have no extra junk mixed in, you’ve achieved 100% precision. But recall and precision are not friends. Typically, a technique that increases one will decrease the other.

This engineering trade-off between recall and precision is why it helps to be explicit and think carefully about what we’re trying to accomplish. Because the three categories of document review have different recall and precision targets, we must choose and tune our technologies—including TAR—with these specific goals in mind so that we maximize effectiveness and minimize cost and risk. Let us explain in more detail.
Classification Tasks

Start with classification: the sorting of documents into buckets. We typically classify so that we can do different things with different subpopulations, such as review, discard or produce.

Under the Federal Rules of Civil Procedure, and as emphasized by The Sedona Conference and any number of court opinions, e-discovery is guided by principles of reasonableness and proportionality. As Magistrate Judge Andrew J. Peck wrote in the seminal case, *Da Silva Moore v. Publicis Groupe*:

*The goal is for the review method to result in higher recall and higher precision than another review method, at a cost proportionate to the value of the case.*

As Judge Peck suggests, when we’re talking document production the goal is to get better results, not perfect results. Given this, you want to achieve reasonably high percentages of recall and precision, but with cost and effort that is proportionate to the case. Thus, a goal of 80% recall—a common TAR target—could well be reasonable when reviewing for responsive documents, especially when current research suggests that the “gold standard” of complete eyes-on review by attorneys can't do any better than that at many times the cost.¹

Precision must also be reasonable, but requesting parties are usually more interested in making sure they get as many responsive documents as possible. So, recall usually gets more attention here.²

Protection Tasks

By contrast, when your task is to protect certain types of confidential information (privilege, trade secrets, confidential information, or anything else where the bell can't be unrung), you need to achieve 100% recall. Period. Nothing can fall through the cracks. This tends to be problematic in practice, as the goal is absolute perfection and the real world seldom obliges.

So to approximate this perfection in practice, we usually need to use every tool in our toolkit to identify the documents that need to
be protected—not just TAR but also keyword searching and human review—and use them effectively against each other. The reason for this is simple: Different review methods lead to different kinds of mistakes. Human reviewers tend to make random mistakes.

TAR systems tend to make systematic errors, getting entire classifications of documents right or wrong. By combining different techniques into our workflows, one serves as a check against the others, ensuring that no relevant documents slip through the cracks.

The best way to maximize recall is to stack techniques.

This is an important point about TAR for data protection tasks, and one we want to reemphasize. The best way to maximize recall is to stack techniques, not replace them. Because TAR doesn’t make the same class of errors as search terms and human review, it makes an excellent addition to privilege and other data protection workflows—provided the technology can deal with low prevalence and be efficiently deployed.

Precision, on the other hand, is somewhat less important when your task is to protect documents. Precision doesn’t need to be perfect, but because these tasks typically use lots of attorney hours, they’re usually the most expensive part of review.

Including unnecessary junk in the review gets expensive quickly. So you still want to achieve a fairly high level of precision (particularly to avoid having to log documents unnecessarily if you are maintaining a privilege log), but recall is still the key metric here.
Knowledge Generation Tasks

The final task we described is where we get the name “discovery” in the first place. What stories do these documents tell? What stories can my opponents tell with these documents? What facts and knowledge can we learn from them?

This is the discovery task that is most Google-like. For knowledge generation, we don’t really care about recall. We don’t need all the documents about a topic; we just want the best documents about a topic—the ones that will end up in front of deponents or used at trial.

Precision is therefore the most important metric here. You don’t want to waste your time going through junk—or even duplicative and less relevant documents. This is where TAR can also help, prioritizing the document population by issue and concentrating the most interesting documents at the top of the list so that attorneys can quickly learn what they need to litigate the case.

But one nitpicky detail about TAR for issue coding and knowledge generation should be mentioned. TAR algorithms rank documents according to their likelihood of getting a thumbs-up or a thumbs-down from a human reviewer. They do not rank documents based on how interesting they are. For example, in a review for responsiveness, some documents could be very easy to predict as being responsive, but not very interesting. On the other hand, some documents could be extremely interesting, but harder to predict because they are so unusual.

On the other hand, however, the more interesting documents tend to cluster near the top of the ranking. Interesting documents sort higher this way because they tend to contain stronger terms and concepts as well as more of them. TAR’s ability to concentrate the interesting documents near the top of a ranked list thus makes it a useful addition to knowledge generation workflows.

What’s Next

With this framework for thinking about, developing, and evaluating different discovery workflows, our next chapter will get into the specifics of how TAR 2.0 can best be used for the various tasks at hand.
Footnotes


2. The differing importance of recall and precision both here and in other discovery tasks is one reason the $F_1$ measure (the harmonic mean of recall and precision) is often problematic. While it may be a good single measure for information retrieval research, it prematurely blends two measures that often have to be considered and weighted separately in practical discovery tasks.

3. See, e.g. Maura R. Grossman and Gordon V. Cormack, Inconsistent Responsiveness Determination in Document Review: Difference of Opinion or Human Error?, 32 Pace L. Rev. 267 (2012), (finding that coding inconsistencies by human reviewers are largely attributable to human error, and not to documents being “borderline” or any inherent ambiguity in the relevance judgments).

4. Random training approaches such as those used by support vector machine algorithms tend to need prohibitively large samples in order to deal effectively with low richness, which is common in many actual cases. See, e.g. Gordon V. Cormack and Maura R. Grossman, Evaluation of Machine-Learning Protocols for Technology-Assisted Review in Electronic Discovery, SIGIR ’14, July 6–11, 2014, Gold Coast, Queensland, Australia (evaluating different approaches to TAR training across eight data sets with prevalence (richness) ranging from 0.25% to 3.92% with a mean of 1.18%).

5. To be more nitpicky, this search is the most Google-like for the basic task of searching on a single topic. A more challenging problem here is often figuring out all the different possible topic that a collection of documents could speak to—including those that we don’t know we need to look for—and then finding the best examples of each topic to review. This is another area where TAR and similar tools that model the entire document set can be useful.

6. This is true in general, but not always. Consider an email between two key custodians who are usually chatty but that reads simply “Call me.” There are no key terms there for a ranking engine based on full text analysis to latch onto, though the unusual email could be susceptible to other forms of outlier detection and search.
In chapter six, we talked about the kinds of tasks lawyers perform in the course of document review: (1) classification, (2) protection and (3) knowledge generation. Because the objectives of protection and knowledge generation tasks differ from the goals of a typical production (classification) review, the continuous active learning (CAL) workflow often differs slightly as well. For any given task, however, there are some fairly standard approaches that work well and take full advantage of the strengths of CAL and the two companion algorithms comprising OpenText™ Insight Predict, contextual diversity and algorithmic QC.

In this chapter, we explore various techniques for implementing a Predict review for purposes other than classification, including:

- Investigations
- Opposing party reviews
- Depo prep and issue analysis
- Privilege and privilege QC
We do so not to limit your creativity; there are a lot of ways to use a CAL predictive ranking algorithm. Rather, it is to get you thinking beyond outbound productions so you can take advantage of the many other things CAL can do.

1. Investigations

Most investigations, whether they are internal investigations, regulatory investigations, or even investigations in anticipation of litigation, are true knowledge generation tasks. The primary objective is to find the critical documents on all the principal issues as quickly as possible. There is no need to find every document—just enough of the key documents to fully understand all of the underlying issues. So, recall is not crucial; precision and coverage are.

And unlike litigation, there are no fact-laced complaints or prescriptive requests for production to focus the search for pertinent documents. An investigation oftentimes begins with, at most, a handful of informative documents, but more often nothing more than vague assertions of some assertedly actionable conduct.

**Single seeds:** The paucity of exemplar documents is not an impediment to an efficient and effective investigation review using CAL. Our studies have shown that even a single positive document can be used to quickly locate the majority of the pertinent documents in a collection.

**Synthetic seeds:** And that document may be a synthetic seed that doesn’t even exist in the subject collection. A synthetic seed is an exemplar document that is created from whole cloth to reflect the key language likely contained in the positive documents within the subject collection. And it can take pretty much any form.

A synthetic seed can be a prose recitation of the facts that are expected to underlie the entire investigation. Or it can simply be a compilation of the keywords (or other features, such as bigrams and trigrams) that are likely to be contained in any positive documents. Whatever form it takes, the document is simply added to the subject collection and marked positive, and a CAL tool will immediately start to elevate similar (likely positive) documents to the top of the ranking for early review.
To illustrate how this can work, we simulated a small investigation to see how quickly Predict could locate the positive documents in a very sparse collection, using only a single document to initiate the ranking. The collection consisted of roughly 4,600 documents, of which only 55 were positive (for a richness of 1.2%). The yield curves for two simulated reviews using (1) a single positive document, and (2) a single synthetic seed are shown below. As the chart illustrates, Predict was able to locate 65% of the positive documents after review of fewer than 130 documents in both cases. That equates to reviewing less than two documents to find each positive document, and a review of less than 3% of the collection.

![Chart showing yield curves for two simulated reviews.]

While every collection differs, the prevailing utility of Predict and CAL to investigations is obvious. Positive documents can be elevated for review very quickly, and with very little information or advance preparation.

There is an additional benefit to using CAL for investigations, that is inherent to the operation of many CAL protocols. CAL is proficient at quickly elevating documents pertaining to the broad range of issues that may be subsumed within an investigation review, without having to focus on any given issue independently. That means that a CAL review will not only elevate positive documents for review quickly, but will do so across a broad spectrum of pertinent issues.
Aspectual Recall Simulations

We simulated this CAL characteristic as well, confirming a study that was first reported in 2015. We evaluated a collection consisting of 521,669 documents that was nearly 42% rich. The collection had been evaluated for responsiveness, as well as thirteen (13) substantive issues, which themselves ranged in richness from 0.13% to 27.7%.

We simulated a responsiveness review, and simultaneously noted the discovery of documents relating to each issue, charting the results as a group of yield curves as shown below. This chart shows that a Predict review for generally positive documents (blue line) will quickly elevate documents covering a wide range of topics at issue (green lines). In this example, by the time the general review had reached 40% of the positive documents, documents for every related issue had been discovered and reviewed.

We saw the same results for documents that had been coded as “hot” (red line) in that collection. In conjunction with the general responsiveness review, we also noted the discovery of hot documents, which we again charted together as yield curves. Consistent with the ability of CAL to elevate documents relating to the various topics at issue in the review, CAL also quickly elevated the important, hot documents for early review and analysis.
The importance of using CAL for investigations to take advantage of this capability is again fairly obvious. The ability to get to the most critical documents first, and to see documents spanning all of the underlying issues, will make an investigation both quick and complete.

**Contextual Diversity**

One of the ancillary benefits of using Predict for an investigation is the ability to continuously search for unknown topics or concepts that might exist in the collection, through the continuous application of contextual diversity, one of the companion algorithms to Predict. As discussed in the next chapter, contextual diversity is constantly ranking the entire collection by how much each document actually differs from what has already been seen and coded.

These “unknown” documents are continually batched for review to ensure an even deeper penetration into the collection than is otherwise attained by the general ability of CAL to ferret out all the peripheral underlying issues. And the ratio of contextual diversity documents in the review batches can be adjusted over time, as the focus of the review moves from a more complete understanding of the known issues, into an exploration of the unknown issues.
Proving a Negative

A corollary to the knowledge generation investigation task is the ability to use Predict and contextual diversity to demonstrate the absence of any documents relating to a particular inquiry—at least to a statistical degree of certainty. This would typically arise in situations such as Supplemental Second Requests, where the body of responsive documents has already been depleted. In that situation, rather than review the entire collection to find nothing, the objective is to review only a sufficient number of documents to adequately demonstrate that there simply are no more responsive documents.

Certainly, Predict and contextual diversity are only two of the weapons in the arsenal needed to “prove a negative,” but they are critical to minimizing the review that will be necessary to achieve appropriate statistical levels.

Since proving a negative is a statistical undertaking, the first step is to set the statistical parameters (confidence level and confidence interval) that will support your conclusion that there are not enough documents in the collection to justify a full review. The confidence level and confidence interval will establish (1) the number of documents that will need to be reviewed and (2) the margin of error for the review. With the margin of error, the maximum number of positive documents in the collection can be estimated, and proportionality considerations can be quantified.

There is no hard and fast rule for setting the statistical parameters for the review, and there are really two approaches. First, setting the size of the ultimate review will establish the statistical parameters—that is, the confidence level and confidence interval. For example, reviewing 5,000 documents will, in every instance, ensure a margin of error of less than 1.4% at a 95% confidence level, and less than 1.9% at a 99% confidence level.

Alternatively, setting the confidence level and confidence interval will establish the size of the review. So, it will take a total of roughly 2,400 documents to ensure a maximum margin of error of 2% at a confidence level of 95%, while it would take roughly 4,200 documents to achieve that same margin of error at a 99% confidence level.4 In either case, the relative cost and benefit of various sample sizes can
be evaluated, and the number of documents to be reviewed can be negotiated and set accordingly.

Before implementing CAL, advanced analytics should be used to review between 20% to 30% of the total number of documents to be reviewed, in an effort to find positive documents. Every reasonable technique should be used: carefully crafted and targeted keyword searches; file type analyses; custodian and timeline analyses; communication analytics, random sampling; etc.

Since no analytics approach is likely to locate any responsive documents (because none are expected to exist in the collection), the entire review should focus on finding documents that are contextually close to being responsive. These “close” documents will eventually serve as the best available training examples for the CAL review.

Once the analytics review is complete, continuous active learning can be used to complete the remainder of the review. The CAL algorithm will efficiently analyze the entire collection to locate any documents that are contextually similar to the “close” documents located during the analytics review, and will continuously learn from every coding decision made along the way.

The full functionality of a CAL tool should be exploited to make every effort to locate responsive documents. One or more synthetic seeds, reflecting the content of a document that would be responsive, should be used to train the tool, along with the “close” documents. Contextually diverse documents should be included, to ensure a thorough exploration of the entire collection. Again, any documents that are contextually close to being responsive should be considered positive, in order to prioritize any truly responsive documents along the way.

Once the remaining documents have been reviewed, and no responsive documents have been found, the underlying statistics can be used to essentially prove a negative—that is, to establish the statistical maximum number of responsive documents in the collection. Proportionality considerations will then determine whether a full review is warranted, just to locate that small number of potentially responsive documents.
Given the effectiveness of a CAL review, this approach is actually more thorough than a traditional random statistical sample review, and is a reasonable way to demonstrate the absence of responsive documents in the collection without having to review the entire collection.

2. Opposing Party Productions

Opposing party productions are essentially knowledge generation tasks as well. The objective is to weed through a collection to find particularly relevant documents. Recall (i.e., finding all of the relevant documents) is not as critical as precision—seeing more relevant documents than irrelevant ones—and surfacing more hot documents in the process.

CAL is particularly suited for this task. First, CAL is efficient in the review of sparse collections. And, despite the general responsiveness of opposing party productions, the truly important documents are few and far between. Second, as discussed earlier, CAL is also a superior way to surface Hot documents along the way.

There are a few different ways to initiate a CAL review of an opposing party production. With the caveat that the language used by opposing parties will typically differ, client documents may provide a reasonable starting point. Relevant opposing party documents provide even better seeds to initiate a CAL ranking. Oftentimes, a handful of such documents are available through past communications, or can be found through a modest analytics assessment of the production—and only a handful of positive documents is enough to start a CAL review. Otherwise, a CAL review can be initiated with a single synthetic seed detailing precisely what is being sought from the opposing party production.

Once the CAL review begins, there is no special workflow needed to effectively review an opposing party production. CAL will elevate relevant documents, including hot documents, and further minimize the number of irrelevant documents that need to be reviewed along the way. And contextual diversity will ensure coverage across the collection.

If, at any point, there is a desire to switch gears and truly focus on
finding hot documents, it’s easy with Predict. Just spin up a new Predict project ranking on the HotDoc field. Every decision to that point will be used to train the CAL algorithm, and Predict will begin to surface hot documents preferentially over even generally responsive documents.

3. Depo Prep and Issue Analysis

Preparing for multiple witness depositions and researching multiple issues are both knowledge generation tasks, and they follow a similar workflow. And both tasks often suffer from low richness within the larger responsive collection, which makes CAL particularly useful.

In both cases, the setup follows the same approach. The typical coding approach is to structure the witness list or issue list as a multi-value field to allow reviewers to select more than one value (witness or issue) for each document. To get even more granular in the coding schema, and even further improve the effectiveness of CAL, each witness or issue can be set up as a separate binary (yes-no) field.

Using either structure, creating a separate Predict project for each issue or witness will ensure multiple simultaneous, independent rankings. That way, ranking and review for every witness and issue will be focused. Each review can then be conducted simultaneously by multiple reviewers, or sequentially by a single reviewer.

This approach will not prevent reviewers from coding the full spectrum of witnesses and issues pertinent to a particular document. Rather, while every reviewer will see documents ranked independently given their specific objective, they will be able to code documents for other issues and witnesses when appropriate. Doing so will correspondingly improve those other rankings.

4. Privilege and Privilege QC

Privilege assessment is a protection task, regardless of whether it is an initial privilege review to locate privileged documents among a group of unreviewed documents slated for production, or a quality control measure to ensure that documents coded as not being privileged are indeed not privileged. In both cases, CAL can be an effective tool in preventing inappropriate production and disclosure.
CAL has its primary utility as an initial privilege review technique when documents are being produced without an eyes-on review—situations such as second requests and subpoenas. In that case, the goal is to effectively locate and withhold all of the privileged documents, without reviewing the bulk of the collection.

Certainly, in most instances, analytics will be used to isolate obviously privileged communications exchanged with counsel. Any privileged documents discovered in this analytics phase can then be used as seed documents to initiate the CAL ranking for further privilege review.

The extent of the effectiveness of a CAL tool during this initial privilege review will then depend largely on the features that are used to inform the CAL algorithm. If email header information (To, From, domains, etc.) is included in the feature set, the CAL algorithm may have the ability to discern the identity of individuals making and breaking privilege, and rank documents for review accordingly.

Otherwise, the algorithm will be constrained to ranking documents based purely on content text. Text-based ranking is critical to an effective privilege review nevertheless, because privileged communications may be subsequently distributed internally, without any reference to counsel. Assuming the general content of the text has been coded as privileged (presumably in the original communication with counsel) a CAL tool will then elevate similar documents as potentially privileged.

Beyond this, the QC algorithms incorporated into a Predict review provide one final defense to privilege disclosure, particularly in a traditional production review. In that situation, every document being produced has been reviewed and coded, inter alia, for privilege. Spinning up a Predict project on privilege, then, will rank the entire collection by the likelihood of each document being privileged. Further, algorithmic QC will rank every document coded as “not privileged” by the likelihood that they are, in fact, privileged. So, the top-ranked documents actually look like they are privileged even though they are coded as “not privileged.” Reviewing the top-ranked documents in this ranking will provide a final measure of assurance that privileged documents are not being produced.
Other Uses

We have no doubt that people will come up with other use cases for CAL-based predictive ranking. We have written about two-tailed reviews, where teams focus on both ends of the ranked spectrum. We also believe CAL-like systems will prove useful for other kinds of searches, including government records inquiries and patent research, with the focus being on using good documents rather than assumed keywords to build out better searches against all kinds of documents.

Footnotes

1. In the simulation, we generated the synthetic seed by excerpting language from a few positive documents, and then compiling the excerpts in a separate, single document. In practice, a synthetic seed would typically be created without reference to known positive documents, as they would serve as seed documents in their own right.

2. This simulation, which actually included an evaluation of the effectiveness of 57 different starting seeds, is described in greater detail in chapter 26: 57 Ways to Leave Your (Linear) Lover.


4. The relationship between statistical parameters and the number of documents reviewed can easily be determined using readily available tools such as those found at www.surveystem.com/sscalc.htm.
Using Contextual Diversity to Find Out What You Don’t Know (and a Bit About Zipf’s Law)

In the early technology-assisted review (TAR 1.0) era, many thought training with randomly selected documents was important to the success of a TAR review. The fear was that attorneys would bias the TAR algorithm if they selected documents for initial training. When challenged, most relied on the old shibboleth: *You don’t know what you don’t know.*

Frankly the “don’t know” point made sense at a time when legal professionals were just realizing that their carefully crafted keyword searches were failing because their targets used different terms. If lawyer keywords couldn’t be trusted, why trust lawyer-selected training seeds? And, random selection was the cornerstone of the TAR 1.0 protocol called simple passive learning. The computer passively (randomly) selected all the training documents.
Then came the research. In 2014, Grossman and Cormack issued their landmark study comparing different TAR protocols. It showed that random selection was the least efficient way to find relevant documents including those you didn’t know about originally. Our research and that of several others consistently told the same tale.

Our goal in this chapter is not to dissect the arguments on either side of the random sampling debate. Rather, we want to focus on contextual diversity, which is an algorithm we created to help find those “You don't know what you don't know” documents.

We also want to have a bit of fun and show you how Zipf’s law (we hadn’t heard of it either) supports our approach and will help you understand why integrating contextual diversity with a continuous active learning (CAL) system is more efficient and effective than random sampling for ensuring topical coverage and avoiding bias.

**What Is Contextual Diversity?**

In a TAR 2.0 CAL system, we continuously use all the judgments of the review teams to make the algorithm smarter (which means you find relevant documents faster). In large part, we feed highly ranked documents to the review team and use their judgments to train the system. However, our continuous learning approach also throws other options into the mix to (1) further improve performance, (2) combat potential bias and (3) ensure complete topical coverage. One of these options that addresses all three concerns is Predict’s contextual diversity algorithm.

Contextual diversity focuses on documents that are highly different from the ones already seen by human reviewers. Because our system ranks all of the documents on a continual basis, we know a lot about the documents—both those the review team has seen and also those the review team has not yet seen. The contextual diversity algorithm identifies documents based on how significant and how different they are from the ones already seen, and then selects training documents that are the most representative of those unseen topics for human review.
It’s important to note that the algorithm doesn’t know what those topics mean or how to rank them. But it can see that these topics need human judgments on them and then selects the most representative documents it can find for the reviewers. This accomplishes two things:

1. It is constantly selecting training documents that will provide the algorithm with the most information possible from one attorney-document view, and

2. It is constantly putting the next biggest “unknown unknown” it can find in front of attorneys so they can judge for themselves whether it is relevant or important to their case.

We feed in enough of the contextual diversity documents to ensure that the review team gets a balanced view of the document population, regardless of how any initial seed documents were selected. But we also want the review team focused on highly relevant documents, not only because this is their ultimate goal, but also because these documents are highly effective at further training the TAR system. Therefore, we want to make the contextual diversity portion of the review as efficient as possible. How we optimize that mix is a trade secret, but the concepts behind contextual diversity and active modeling of the entire document population are explained below.

**Contextual Diversity: Explicitly Modeling the Unknown**

In the following example, assume you started the training with contract documents found either through keyword search or witness interviews. You might see terms like the ones above the blue dotted line showing up in the documents. Documents 10 and 11 have human judgments on them (indicated in red and green), so the TAR system can assign weights to the contract terms (indicated in dark blue).
But what if there are other documents in the collection, like those shown below the dotted line, that have highly technical terms but few or none of the contract terms? Maybe they just arrived in a rolling collection. Or maybe they were there all along but no one knew to look for them. How would you find them based on your initial terms? That’s the essence of the bias argument.

With contextual diversity, we analyze all of the documents. Again, we’re not solving the strong artificial intelligence problem here, but the machine can still plainly see that there is a pocket of different, unjudged documents there. It can also see that one document in particular, 1781, is the most representative of all those documents, being at the center of the web of connections among the unjudged terms and unjudged documents.

Our contextual diversity engine would therefore select that one for review, not only because it gives the best “bang for the buck” for a single human judgment, but also because it gives the attorneys the most representative and efficient look into that topic that the machine can find.
So Who Is This Fellow Named Zipf?

Zipf's law was named after the famed American linguist George Kingsley Zipf, who died in 1950. The law refers to the fact that many types of data, including city populations and a host of other things studied in the physical and social sciences, seem to follow a Zipfian distribution, which is part of a larger family of power law probability distributions. (You can read all about Zipf's law in Wikipedia, where we pulled this description.)

Why does this matter? Bear with us, you will see the fun of this in just a minute.

It turns out that the frequency of words and many other features in a body of text tend to follow a Zipfian power law distribution. For example, you can expect the most frequent word in a large population to be twice as frequent as the second most common word, three times as frequent as the third most common word and so on down the line. Studies of Wikipedia itself have found that the most common word, "the," is twice as frequent as the next, "of," with the third most frequent word being "and." You can see how the frequency drops here:
Topical Coverage and Zipf’s Law

Here’s something that may sound familiar: Ever seen a document population where documents about one topic were pretty common, and then those about another topic were somewhat less common, and so forth down to a bunch of small, random stuff? We can model the distribution of subtopics in a document collection using Zipf’s law too. And doing so makes it easier to see why active modeling and contextual diversity is both more efficient and more effective than random sampling.

Here is a model of our document collection, broken out by subtopics. The subtopics are shown as bubbles, scaled so that their areas follow a Zipfian distribution. The biggest bubble represents the most prevalent subtopic, while the smaller bubbles reflect increasingly less frequent subtopics in the documents.

Now to be nitpicky, this is an oversimplification. Subtopics are not always discrete, boundaries are not precise, and the modeling is much too complex to show accurately in two dimensions. But this approximation makes it easier to see the main points.

So let’s start by taking a random sample across the documents, both to start training a TAR engine and also to see what stories the collection can tell us.

We’ll assume that the documents are distributed randomly in this population, so we can draw a grid across the model to represent a simple random sample. The red dots reflect each of 80 sample documents. The portion of the grid outside the circle is ignored.
We can now represent our topical coverage by shading the circles covered by the random sample.

You can see that a number of the randomly sampled documents hit the same topical circles. In fact, over a third (32 out of 80) fall in the largest subtopic. A full dozen are in the next largest. Others hit some of the smaller circles, which is a good thing, and we can see that we've colored a good proportion of our model yellow with this sample.

So in this case, a random sample gives fairly decent results without having to do any analysis or modeling of the entire document population. But it's not great. And with respect to topical coverage, it's not exactly unbiased, either. The biggest topics have a ton of representation, a few tiny ones are now represented by a full 1/80 of the sample, and many larger ones were completely missed.

So a random sample has some built-in topical bias that varies randomly—a different random sample might have biases in different directions. Sure, it gives you some rough statistics on what is more or less common in the collection, but both attorneys and TAR engines usually care more about what is in the collection rather than how frequently it appears.

What if we actually can perform analysis and modeling of the entire document population? Can we do better than a random sample? Yes, as it turns out, and by quite a bit.

Let's attack the problem again by putting attorney eyes on 80 documents—the exact same effort as before—but this time we select the sample documents using a contextual diversity process. Remember: our mission is to find representative documents from as many topical groupings as possible to train the TAR engine most
effectively, avoid any bias that might arise from judgmental sampling, and to help the attorneys quickly learn everything they need to from the collection. Here is the topical coverage achieved using contextual diversity for the same size review set of 80 documents:

Now look at how much of that collection is colored yellow. By actively modeling the whole collection, the TAR engine with contextual diversity uses everything it can see in the collection to give reviewing attorneys the most representative document it can find from each subtopic. By using its knowledge of the documents to systematically work through the subtopics, it avoids massively oversampling the larger ones and having to rely on random samples to eventually hit all the smaller ones (which, given the nature of random samples, need to be very large to have a decent chance of hitting all the small stuff). It achieves much broader coverage for the exact same effort.
At right is a comparison of the two different approaches to selecting a sample of 80 documents. The subtopics colored yellow were covered by both. Orange indicates those that were found using contextual diversity but missed by the random sample of the same size. Dark blue shows those smaller topics that the random sample hit but contextual diversity did not reach in the first 80 seed documents.

Finally, here is a side by side comparison of the topical coverage achieved for the same amount of review effort:

Now imagine that the attorneys started with some judgmental seeds taken from one or two topics. You can also see how contextual diversity would help balance the training set and keep the TAR engine from running too far down only one or two paths at the beginning of the review by methodically giving attorneys new, alternative topics to evaluate.

When subtopics roughly follow a Zipfian distribution, we can easily see how simple random sampling tends to produce inferior results.
compared to an active learning approach like contextual diversity. In fact, systematic modeling of the collection and algorithmic selection of training documents beats random sampling even if every topic were the exact same size, but for other reasons we will not go into here.

The goal in e-discovery review is to find relevant documents as quickly and efficiently as possible while also helping attorneys learn everything they need to know about to litigate the case effectively. With contextual diversity, George Zipf is in our corner.

---

Footnotes

In the early days, many questioned whether technology-assisted review (TAR) would work for non-English documents. There were a number of reasons for this but one fear was that TAR only “understood” the English language.

Ironically, that was true in a way for the early days of e-discovery. At the time, most litigation support systems were built for ASCII text. The indexing and search software didn’t understand Asian character combinations and thus couldn’t recognize which characters should be grouped together in order to index them properly. In English (and most other Western languages) we have spaces between words, but there are no such obvious markers in many Asian languages to denote which characters go together to form useful units of meaning (equivalent to English words).

Over time, litigation support systems advanced and added the capability to recognize different languages, handle Unicode rather than ASCII text and, ultimately, to properly search words or phrases in the challenging CJK languages (Chinese, Japanese and Korean) as well as other non-Western languages such as Arabic, Hebrew and
those in the Cyrillic alphabets. Catalyst, for example, upgraded its search system in 2008 to handle multi-language discovery.

We faced the same issues when the first TAR 1.0 systems hit the market in the 2010s. They, like many of their underlying search systems, were still not built to handle the more challenging Asian and Eastern languages and many questioned whether they had any utility for foreign-language discovery. Indeed, in 2014, the U.S. Department Of Justice published a memo that talked about the importance of TAR for their investigations but expressed skepticism about the use of TAR for mixed or non-English language collections.

**TAR 2.0 and Multi-Language Discovery**

All that changed in the TAR 2.0 era. To understand why TAR can work with non-English documents, you need to know two basic points:

- **TAR doesn’t understand English or any other language.** It uses an algorithm to associate words/tokens with relevant or irrelevant documents.

- **To use the process for non-English documents, particularly those in Chinese and Japanese, the TAR system has to first tokenize the document text so it can identify individual words.**

We will hit these topics in order.

**1. TAR doesn’t understand English.**

It is beyond the province of this chapter to provide a detailed explanation of how TAR algorithms work, but a basic explanation will suffice for our purposes. Let me start with this: the TAR algorithm doesn’t understand English or the actual meaning of documents. Rather, it simply analyzes words (tokens) according to their frequency in relevant documents compared to their frequency in irrelevant documents.

Think of it this way: We train the system by marking documents as relevant or irrelevant. When we mark a document relevant, the computer algorithm analyzes the words in that document and ranks
them based on frequency, proximity or some other such basis. When we mark a document non-relevant, the algorithm does the same, this time giving the words a negative score. At the end of the training process, the computer sums up the analysis from the individual training documents and uses that information to build a model it can use to analyze a larger set of documents.

While algorithms work differently, think of a TAR system as creating searches using the words developed during training. Only some of the words will have a positive weight and some of the words will have a negative weight. Documents responding to the search are then returned in an ordered ranking, with the most likely relevant ones coming first.

None of this requires that the computer know the English language or understand anything about the meaning of the returned documents. All the computer needs to know is which words are in which documents and how often they appear.

2. If documents are properly tokenized, the TAR process will work best.

Tokenization may be an unfamiliar term to many but it is not difficult to understand. When a computer processes documents for search, it pulls out all of the words and places them in a combined index. When you run a search, the computer doesn't go through all of your documents one by one. Rather, it goes to an ordered index of terms to find out which documents contain which terms. That's why search works so quickly. Even Google works this way, using huge indexes of words.

As we mentioned, however, the computer doesn't understand words or even that a word is a word. Rather, for English documents it identifies a word as a series of characters separated by spaces or punctuation marks. Thus, it recognizes the words in this sentence because each has a space (or a comma or a period) before and after it. Since not every group of characters is necessarily an actual “word,” information retrieval scientists call these groupings “tokens.” They call the act of recognizing these tokens for inclusion in an index as “tokenization.”
All of these are tokens:

- Bank
- door
- 12345J
- barnyard
- mixxpelling

If they are separated by a space or punctuation, they will be recognized by the indexer and cataloged as tokens (words or otherwise) in the index.

Certain languages, such as Chinese, Japanese and Korean, don’t delineate words with spaces or Western punctuation. Rather, they string individual characters together, often with no breaks at all. It is up to the reader to group characters into words or phrases in order to understand their meaning.

Here is an example:

**I do not know Japanese.**

In English it is easy to distinguish the words (tokens) because they are separated by spaces.

With this same sentence in Japanese, it is much more difficult to determine where individual words begin and end.

にほんごをはなしません。

Special tokenization software is required to determine what language is being used (Japanese in this case) and which characters should be grouped together to form what we think of as “words” in English.

Properly tokenized, the Japanese sentence would look this way:

にほんご を はなし ません。
And tokenize it thusly:

にほんご
を
はなし
ません

It is easy to imagine that a good TAR system can find relevant documents more efficiently when the words or phrases are properly tokenized.

**TAR 1.0 Tokenization**

TAR 1.0 systems were focused on English-language documents and could not tokenize Asian text. As a result, they defaulted to treating each character as a token, or taking the whole line of text (which could include parts of several sentences) as a token. Using English for analogy it would be like indexing the sentence above (“I do not know Japanese”) using the following tokens with their respective counts:

```
   i 1
d 1
o 3
n 3
t 1
k 1
w 1
j 1
a 2
p 1
e 2
s 1
```

Maybe that approach could surface some relevant documents based on character overlap alone. However, you can be certain the system will not be as effective using this approach.

At Catalyst, we added special language identification and tokenization software to make sure that we handle these languages properly. As a result, our TAR system can analyze Chinese and Japanese documents just as well as English documents. Word frequency counts are just as effective for these documents and the resulting rankings are as effective as well.
Conclusion

As corporations grow globally, legal matters are increasingly likely to involve non-English language documents. Many believed that TAR was not up to the task of analyzing non-English documents. The truth, however, is that with the proper technology and expertise, TAR can be used with any language, even historically difficult languages such as Chinese and Japanese.

We know from experience that a TAR 2.0 system with proper language identification (required to know how to tokenize) and a strong tokenization library can be as effective with non-English languages as with English. To go further on this subject, read the case study on *Using TAR 2.0 to Streamline Document Review for Japanese Patent Litigation* later in this book. And we suggest you read our case study called *57 Ways to Leave Your Linear Lover*, which also focuses on Japanese documents.

Whether for English or non-English documents, the benefits of TAR are the same. By using computer algorithms to rank documents by relevance, lawyers can review the most important documents first, review far fewer documents overall, and ultimately cut both the cost and time of review. In the end, that is something their clients will understand, no matter what language they speak.

Footnotes

1. There is another approach to tokenization that can be language independent. The tokenizer takes, for example, the first three characters and makes them a token. It then takes characters 2 through 4 and makes that a second token. Then 3 through 5 and so on. These tokens are then compared for frequency much like individual words might. However, we know of only one group using this approach in the legal industry. Many others are still using the TAR 1.0 approach.
What Are the Thresholds for Using Technology-Assisted Review?

A common question among legal professionals is whether there is a minimum threshold (in terms of numbers of documents) before a technology-assisted review (TAR) makes sense. If so, what is that threshold?

This has been a hot topic since the advent of TAR. Many TAR 1.0 users took the position that TAR was only for larger matters, with many putting the threshold at 100,000 documents. Others suggested it might still be useful in smaller cases, but they were talking about cases with 50,000 documents. Nobody suggested TAR would work for cases smaller than that.

The reason lies in the training overhead required for the TAR 1.0 protocol. TAR 1.0 required that a SME hand code 500 documents to act as a reference set, then review and tag another 1,500 to 3,000 documents to train the algorithm. And then review another 500 or so documents for validation QC. All of this had to happen before review could begin.
TAR 2.0 proponents offer a different view on this question. The short answer is that there are no size thresholds for a properly built TAR 2.0 system. It is as effective with small collections as with large ones. The simple reason is that there is little or no training overhead in a TAR 2.0 process.

**TAR 1.0**

The TAR 1.0 process is divided into two sequential stages: (1) training and (2) review. Proponents of TAR 1.0 have recommended for years that only senior attorneys or experts be allowed to judge any document that would be used for training, a refrain that was repeated even at the Stanford CodeX FutureLaw Conference. If this advice is to be followed, it stands to reason that, even if training sets are of practical necessity, they are going to be limited to no more than a few thousand documents. After all, what senior attorney wants to sit there and review for more than three or four solid days in a row?

This one-two punch of requiring experts to train the system and separating the workflow into two distinct phases (training followed by review) carries with it two implications. First, a certain minimum amount of training needs to happen and a certain maximum amount of training will be able to happen. Second, a certain minimum number of relevant documents need to be found within that limited training set in order for the system to be properly trained.

Think about what this means.

One consequence is that, because of the separate training phase, the collection needs to be a certain size in order for TAR to be worth using. For example, a common training set size is around 2,000 documents. If the entire collection is only 2,500 documents, and 2,000 documents of training are required before a single prediction kicks in, then the “technology assisted” portion of the review applies to only 500 documents.

It is very possible that it will cost you more to train your TAR algorithm using your experts (not even counting the cost of whatever portion of the remaining 500 still need to be reviewed) than it would have cost to just send all 2,500 documents to a contract review team to review in an old-fashioned linear manner.
Thus, there is a calculation that the e-discovery practitioner has to make when trying to figure out how much it will cost to train a TAR 1.0 system versus how much benefit can be derived from it.

Where is that optimal balance point, the optimal document collection size at which it makes economic sense to bring in a TAR 1.0 system? Is it with collections that are 2,500 documents in size? Five thousand? Twenty thousand? One hundred thousand? In truth, vendors who recommend this workflow can't always rightly say what that optimal minimum size is going to be, because the answer isn't only size dependent, it's also data dependent. We have heard these vendors say that the collection size should be “large enough.”

The other consequence of the TAR 1.0 workflow is that if the collection is not rich enough in relevant material, and your machine learning classifier requires a certain number of positive documents in order to be properly trained, you will never find enough positive examples in the limited training set to properly train your classifier.

Case in point: One vendor claims that its system needs only 300 (rather than 1,500 or even 2,000) training documents in order to start making predictions, but in those 300 documents there have to be at least 50 positive examples. We handled a small matter recently in which there were only 41 positive documents in the entire collection of about 15,000 documents. With such an extremely low richness (41/15000 = 0.27%), the vendor that requires a certain minimum number of relevant documents would never be able to make a single prediction. Users of that system would have to review all 15,000 documents.

In contrast, a TAR 2.0 system like Insight Predict found 40 of the 41 positive documents after reviewing approximately 800 of the 15,000 documents in the collection. Forty documents out of 800 is not a terribly high precision rate on this particular matter—most cases we deal with have a much higher precision—but it is certainly an order of magnitude better than the realistic alternative of having to look at all 15,000 to find those 40 documents.

Thus, the question about whether TAR only works with a certain minimum collection size or for a certain richness has a lot to do with the native design of the algorithm and the process or workflow.
around that algorithm. The decision that virtually all vendors have made for years—to separate training and review—has real consequences for the collection size at which TAR makes economic sense. Further, the decision that virtually all vendors have made for years to require that any document that is used as a training document has to be judged by the expert or senior attorney also has practical consequences for how large the training set can be, which in turn affects which collections work or do not work for TAR.

Consequently, an entire generation of TAR users has been trained to wait until the “right case” comes along to use TAR. Technology adoption in the legal space is already painfully slow as is. It did not help that too many vendors slowed it down even further with these impractical and limiting workflow recommendations.

With Insight Predict, on the other hand, we designed our algorithms and processes from the ground up to be both continuous and robust to non-expert, non-senior attorney judgments. With Predict, training is review and review is training; they are not two separate stages. They are integrated into a single, holistic, continuous stage.

As a consequence of Predict's unique design, there is no minimum collection size and no minimum collection richness needed for the predictive technology to work.

With Predict, you hit the ground running. I've seen our system work on collections not just of millions or hundreds of thousands of documents, but on 15,000 documents or 8,000 documents. We've even done simulated trials on collections of 400 documents. It works on all of them. We've seen our system work on matters with 80% richness and matters with less than 0.5% (that's one-half of one percent) richness. We have seen Predict work when the initial documents fed to the system were in the thousands (via an export from another vendor when the client decided to switch technology providers midstream), and we have seen Predict work when only a single document has been fed to the system.

This non-thresholding approach opens up a new world of possibilities. For example, we've been using Predict for years now to do things such as depo prep, whereby you quickly spin up a Predict project for a deponent you’re going to interview tomorrow, and start
getting predictions on that deponent’s documents after coding only a single training document. And you can be done after looking at just a few hundred documents, total, because you’ve already found the information you needed. Most, if not all, other systems haven’t even finished their training, much less found everything they needed to find, after looking at a few hundred documents.

So as noted at the outset, the answer to the question of how many documents you need in your matter for TAR to be effective depends on the type of TAR system you are using. For TAR 1.0, the number is high, with a heavy dependence on the richness of the documents being processed. With a TAR 2.0 process, then number can even be in the hundreds. In fact, we recommend that TAR 2.0 be used in every case because it quickly moves relevant documents to the front of the queue, no matter what type of document collection you have.
How Good Is That Keyword Search? Maybe Not as Good as You Think

Despite advances in machine learning over the past half-decade, many lawyers still use keyword search as their primary tool to find relevant documents. Keyword searches can be an easy and effective way to isolate a small set of documents that are related to a specific topic. But trying to use them as a primary search method or as a culling mechanism to pare down a collection to a smaller review set can be ineffective, and in some cases, downright reckless. There are some search experts in the field who are able to achieve good results with complex keyword searches, but proficiency in this task requires tremendous skill and years of specialized training to develop, which most of us simply don’t have.

Many e-discovery protocols are built around reaching agreement on keywords, but few require testing to see whether the keywords are missing large numbers of relevant documents. Rather, many seem to believe that if they frame the keywords broadly enough they will find most of the relevant documents, even if the team is forced to review a lot of irrelevant ones.
In nearly every instance, continuous active learning (CAL) is far more effective than keyword search at finding relevant material while keeping irrelevant results to a minimum. And with advanced CAL systems such as Predict, there is usually little need to cull at all. Sure, if your collection winds up being several terabytes, some sort of culling makes sense to get it down to a more manageable size. And if the review protocol is limited to certain dates or custodians, culling the collection by the appropriate metadata values will remove extraneous data from the review collection. But if your goal is to reduce the number of documents that your team will need to review, using CAL on a larger data set will be both more efficient and more defensible than relying on keyword culling.

**Precision vs. Recall: It Can Be Deceiving**

Using terms borrowed from the science of information retrieval, keyword advocates believe they are achieving high recall (the percentage of relevant documents found) while hoping that precision (the number of relevant versus total documents) also stays high. Most know there is a tradeoff between recall and precision—the better the recall, the lower the precision. Even more important, the opposite is often true—better precision almost always means worse recall. As a result, when the keywords seem to bring back a lot of relevant documents (i.e. the search is precise), they become convinced they have found most of the relevant documents in the bargain (i.e. they also have high recall).

Is that true? Most search experts would say no. If your search goal is to find a nearby Italian restaurant and your search brings back several relevant options, that feels like a good result. However, if your goal is to find everything responsive to your search, you might not have such a good result if there are many local Italian restaurants that your search did not return. Precision doesn’t often come with high recall.

**Why Do We Care About Precision and Recall?**

Precision is important in a review because higher precision means fewer non-relevant documents will be reviewed. And fewer irrelevant documents in your review leads to savings in time and
money. But recall is the metric to really pay attention to, since it measures the completeness of the review. If you don’t have high enough recall (generally at least 75%), your review might not meet acceptable standards. So what we’re looking for is a method that will achieve good precision to cut down on wasted effort (i.e. looking at irrelevant documents), while also achieving high recall to ensure the completeness of the review.

How Efficient Are Keywords?

This question might be tougher to answer than you think. While keyword searches can be effective in finding some relevant documents, the process rarely achieves a sufficient level of recall for a true comparison with a CAL process.

Anecdotally, we have asked a number of e-discovery professionals how efficient their keyword-based reviews typically are. While we aren’t claiming this was a statistical survey, the typical answer we received was 10 to one. That means that the typical reviewer has to look at 10 documents for every relevant one found, which equates to 10% precision. We often refer to this metric as the “efficiency ratio,” which in this case would be 10:1. That is a pretty low efficiency rate in our view. If these estimates are anywhere near correct, that suggests that a CAL process like Insight Predict will be about 500% more efficient than keyword review. Put another way, if you are basing your review on keyword search, you are reviewing too many documents.

To be sure, keyword search still has a place in e-discovery, to find relevant documents quickly or when searching for documents that you know have specific and not often-repeated terms. But when it comes to a general review, here’s what the data shows about keyword efficiency—and why Predict is a superior method for document review.

Apples to Apples: The Recall-Precision Tradeoff in Keyword Searching

There is not a lot of publicly available data on the true effectiveness of keyword search, but here is what we do know about keyword
search efficiency. There will always be a tradeoff between recall and precision (or, in this case, efficiency). As a general matter, you can only increase recall levels by sacrificing precision. With that in mind, we can take a look at the keyword search data and understand, at least qualitatively, how to compare keyword search to TAR, given that TAR typically attains much higher recall levels than keyword search.

**Blair-Maron:** Even though it’s over 30 years old, the Blair-Maron study remains the most comprehensive analysis of keyword effectiveness in the legal realm. A team of lawyers conducted 51 different information requests for the study. In each case they generated keyword searches that they believed would be effective.

Each attorney was asked to consider the result set and determine whether it met the initial recall criteria of 75%. If not, and that was the norm for early rounds, the attorneys were able to continue revising the query until they were satisfied with the results. As Blair and Maron stated:

> In the test, each query required a number of revisions, and the lawyers were not generally satisfied until many retrieved sets of documents had been generated and evaluated.

So, how did they do? The lawyers thought it went great. They were convinced they had found 75% of the relevant documents for each of the 51 requests. The research showed otherwise. In three requests, the lawyers found at least 50% of the relevant documents. For the remaining 48 requests, the numbers were much lower, dropping to as low as 4% for a few. The average recall was 20% with the average precision of the searches standing at 80%.

In essence, the lawyers were satisfied because they found a lot of relevant documents in their results set. However, they were fooled because those documents represented only a small fraction of the relevant population. So while attorneys were able to use keyword search to find a lot of relevant documents (indeed, sometimes achieving 80% precision on individual topics), they were only able to find about 20% of the total relevant ones in the larger population. The conclusion which follows logically is that had the attorneys tried to develop additional keywords to increase total recall from 20% to, say, 75% or 80%, they would have had to review a huge number of non-relevant documents in the bargain.
In the end, Blair and Maron concluded that keyword search would never be effective for a large document collection, which in this case was only 40,000 documents. There have been some objections raised to the Blair-Maron results mainly due to the outdated technology available at the time, but their results are consistent with the majority of what we currently see in the field.

**Biomet:** We do have one other publicly available data point that we might use to further evaluate the recall-precision tradeoff for keyword search, this time focusing on keyword culling. In the Biomet matter (Biomet M2a Magnum Hip Implants Prods. Liab. Litigation, N.D. Ind. April 18, 2013), the producing party used keywords to reduce the 19.5 million collected documents to a review set of 2.5 million documents.

The reported statistics in the Biomet matter are not entirely consistent, but do provide a reasonable basis for assessment. Using one set of statistics, we can estimate the recall of the keyword search to be roughly 60%, which means the parties only found 60% of the relevant documents in the entire population.

What about the precision or efficiency of their review? Based on information disclosed in the opinion, we calculate that the precision of their keyword search efforts was roughly 16% (an efficiency of 6.25:1). Using a second set of statistics from the case, the precision would only be about 9% (11:1 efficiency). Neither figure seems to account for the fact that the team likely reviewed many non-responsive family members, so we need to treat the Biomet precision values conservatively. Their actual efficiency may well have been worse than 11 to 1. For comparison, Predict has an average efficiency rate of 1.74:1 at the 60% recall level.

Since neither the Blair-Maron study nor the Biomet decision reflect the higher levels of recall regularly seen with CAL (for example, we typically set our recall target at 80%), we need to consider the recall-precision tradeoff. Obviously, at 80% recall, the precision of keyword searching would be nowhere near the 80% level seen with the Blair-Maron study. Instead, keyword search precision at 80% recall would more likely be nearer (and probably less than) the 16% precision calculated in Biomet. From these figures, we can estimate the precision of keyword search at 80% recall to be roughly 10%, which equates to a 10:1 efficiency.
Catalyst Case Study: Our client collected over 1.4 million documents and loaded them into Insight. Using keyword terms agreed upon by both sides, we culled the population down to around 250,000 documents. These documents comprised the initial review set. The thought, at least from those who created the keyword set, was that most of the relevant documents would be found here.

Our team was concerned (rightly so it turned out) about leaving so many documents out of the review population. What did the keyword searches leave out? What percentage of relevant documents (recall) did the keyword searches actually deliver?

To find out, we pulled a random sample from this “discard pile.” That sample turned out to be 36% responsive, which means that over 420,000 responsive documents did not even make it into the review set. The number of responsive documents left behind would have been nearly double the size of the entire keyword-culled review population.

Put another way, even if we assumed that all of the keyword hits were responsive (i.e. that they achieved 100% precision, which was not the case, as the precision turned out to be around 60%), the keyword searches found only about 38% of the responsive documents. That doesn’t seem adequate by anyone’s standards.

In light of this alarming information, the client decided to add most of the discard pile back into the review set. They obviously needed to review many more documents than they had originally planned, but the review was defensible as well as efficient using Predict.

Predict: Achieving High Efficiency and High Recall

We have found that, on average, a Predict review has the potential to reach a 1.75:1 efficiency ratio at 75% recall. This means, in the perfect world of a simulation, you would have to review 1.75 documents to find each relevant document during the course of review. This is the equivalent of 57% precision.

We then took a look at how those same cases played out in the real world by looking at the statistics for the actual review in its
entirety, including all documents reviewed for the project such as family members, samples, and other document sets that were not prioritized by Predict. On average, the efficiency of all reviews was 2.66:1. But the average recall in those cases was in the neighborhood of 90%—which is generally higher than necessary in litigation, and certainly higher than you would see with keyword search.

Looking at the statistics at the point in which those same cases achieved 80% recall (a more realistic litigation target), the efficiency increases to about 2:1, or 50% precision. This means that you will look at two documents in a typical Predict review to find each responsive document. So if you need to find about 10% of a collection to reach your estimated recall goal, you’ll wind up looking at about 20% of the collection during the course of the review.

**Putting It All Together**

We’ve seen how keyword searches can be very precise, but that comes at the expense of recall. You simply aren’t going to return enough relevant material with most keyword searches while maintaining decent precision. If you try to broaden your searches to return more documents, that comes at the expense of precision and you will likely wind up with a result set that is largely non-relevant. It’s a constant give-and-take that very few search experts can balance in the best of times.

To get a better sense of the overall effectiveness of keyword search, the first chart on the following page shows the precision and recall for the keyword searches in the Blair & Maron study. The goal is to get to the upper right of the chart, as close to 100% precision and recall as possible. And while many of the keyword searches attained high precision (up high on the y-axis), very few even get close to 50% recall, as they’re mostly concentrated around the 20% recall level. And no review is going to be considered anywhere close to complete with 20% recall.
CAL, on the other hand, gets it right. Even while maintaining good levels of precision, Predict will achieve levels of recall that are simply unattainable by keyword searches. This balance of precision and recall is exactly what Predict was designed to do.
The Bottom Line

Whether you are using keyword searching alone or in some combination with machine learning, your goal should be to get it right. No matter how carefully you craft your search terms, keyword search for document review is largely ineffective. The only way to be sure your searches are sufficiently comprehensive is by testing your results. Take a random sample from both the culled set and the discard pile to estimate how many responsive documents are in each group. And if the results leave you feeling uneasy, you might want to think twice about keyword culling.

As we've shown (at least by these examples), using an advanced CAL system such as Predict will almost always outperform keyword searches for any purpose. Simply let the system prioritize the review for you and allow machine learning to do what it's designed to do, which is make correlations between text and responsiveness by utilizing the coding that the reviewers are putting into the system all day long. Not only will your review be more efficient and complete, you'll have more time to spend making higher-level decisions and won't be wasting time and money crafting keyword searches and arguing with the other side over their implementation. Those keyword searches probably aren't doing you a whole lot of good anyway.
In chapter four we asked the question: How much you can save with the second generation of technology-assisted review (TAR) with continuous active learning (CAL)? The discussion in that chapter was based on Maura Grossman and Gordon Cormack’s landmark study on TAR protocols. In this chapter we report on some of the work we have done to compare the efficiency of TAR 1.0 and 2.0 using review simulations, giving TAR 1.0 the benefit of several levels of training for the comparison.

**The Simulation**

The question we answer in this simulation is whether it is more effective to conduct a simple learning (SAL or TAR 1.0) review trained using experts or a continuous learning (CAL or TAR 2.0) review trained using all available reviewers.

The TAR 1.0 approach uses a protocol where the highest skilled or most knowledgeable reviewers train the system for a finite amount of time, followed by a batched-out review in which no machine learning
takes place and the entire review team reviews documents in the order that is predicted by the machine learning algorithm as it was trained by the expert reviewers.

The TAR 2.0 approach uses all available reviewers, begins ranking and re-ranking the moment the first document is judged by any reviewer, and does not cease re-ranking until the target recall point is hit.

The basic parameters of the experiment are as follows:

<table>
<thead>
<tr>
<th>TAR TYPE</th>
<th>TAR 1.0</th>
<th>TAR 2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seed Documents</td>
<td>660</td>
<td>660</td>
</tr>
<tr>
<td>Update Rate</td>
<td>685</td>
<td>685</td>
</tr>
<tr>
<td>Feature Extraction</td>
<td>[same]</td>
<td>[same]</td>
</tr>
<tr>
<td>Algorithm</td>
<td>[same]</td>
<td>[same]</td>
</tr>
<tr>
<td>Training</td>
<td>Experts</td>
<td>Everyone</td>
</tr>
</tbody>
</table>

Charting the Results

Figure 1 shows the main result. The TAR 2.0 line is shown in blue. TAR 1.0 lines are shown in a fade from red to green, with each line reflecting greater levels of training. That is, because different levels of TAR 1.0 training could be done, we show the gain curve after approximately every 3,600 documents of training.

However, because this information is so dense, we also show the same information broken out into three separate graphs. In Figure 2, we show training after approximately 3,600, 7,200 and 10,800 documents. In Figure 3, we show training after approximately 14,400, 18,000 and 21,600 documents. And in Figure 4, we show training after approximately 25,200, 28,800, 32,400, 36,000 and 39,600 documents.
Figure 1: TAR 2.0 (blue) vs. TAR 1.0 (red through green). Theoretical perfect (dotted blue) and manual linear reviews (dotted black) also shown.

Figure 2: TAR 2.0 (blue) vs. TAR 1.0 (red through green) gain curves for 3,600, 7,200 and 10,800 documents of training.
Figure 3: TAR 2.0 (blue) vs. TAR 1.0 (red through green) gain curves for 14,400, 18,000 and 21,600 documents of training.

Figure 4: TAR 2.0 (blue) vs. TAR 1.0 (red through green) gain curves for 25,200, 28,800, 32,400, 36,000 and 39,600 documents

Analysis

Figure 1 shows that while TAR 1.0 can get close to TAR 2.0 at certain levels of recall, at no point does it outperform it. Of course, the answer is not quite as simple as just saying that one approach is better. Take, for example, the first (reddest) TAR 1.0 gain curve in Figure 3, which was produced after training for approximately
14,400 documents. If the stopping point is 85% recall (halfway between 2.39 and 2.69 on the y-axis), then there is practically no difference between the TAR 1.0 and the TAR 2.0 approach. However, if the stopping point is 90% recall (2.69 on the y-axis), then the TAR 2.0 approach beats the expert approach by approximately 28,000 documents.

Or take the gain curves on Figure 4. These show that with enough expert training, the TAR 1.0 approach gets to 94–96% recall at about the same point as the TAR 2.0 approach. However, if the target is 80% recall, the TAR 2.0 approach beats the TAR 1.0 approach by approximately 17,000 documents.

Therefore, in addition to presenting the raw results, we would like to do a cursory analysis of some of the factors that go in to interpreting these results. While a full discussion of these factors is beyond the scope of this write-up, certain general observations can be made.

The three factors that we believe should go into a full analysis of these results are:

1. Knowing when to stop
2. The cost of using the expert
3. The time it takes to execute a simple learning protocol

Which of these factors is most important might change from matter to matter. Sometimes time is of the essence; sometimes cost is more important. The goal of this report is simply to raise awareness of the effect of TAR 1.0 vs. TAR 2.0 protocols on these factors.

**Knowing When to Stop**

The first problem of an expert-trained, TAR 1.0 protocol is knowing when to stop. As Figures 1 through 4 show, the overall gain curve is sensitive to the stopping point. Stop training too early (Figure 2) and it will take much longer to get to high recall. Stop training too late (Figure 4) and you’ll more quickly get to high recall after that point, but you will have done so much training that your overall review effort (and therefore cost) is still greater than it should be.

The problem is hitting that sweet spot of exactly the right amount of training. We are not going to delve into all the factors that need to be
taken into account to decide how much training to do, but the point is that it remains a challenge for TAR 1.0 systems. In general, the fewer critical decisions that have to be made, the better. The TAR 2.0 approach requires only one critical decision: when to stop reviewing. But the TAR 1.0 approach requires two critical decisions: (1) when to stop training and then (2) when to stop reviewing. While it is not impossible to get both decisions correct, it is much more difficult than getting just one decision correct. Figures 5 through 8 show the consequences of incorrectly deciding the stopping point for training.

**Cost of Using the Expert**

In order to do a comparison between TAR 1.0 and TAR 2.0, we must select one of the TAR 1.0 gain curves from Figure 1 as the basis of the comparison. Arguably, the best among these curves is the one that was produced using approximately 14,400 expert training documents. We reproduce this curve in Figure 5.

![Figure 5: TAR 2.0 (blue) vs. TAR 1.0 (red) gain curves for 14,400 documents of TAR 1.0 training](image)

Here again, even though this is arguably the best TAR 1.0 curve, the TAR 2.0 curve beats it at all points. At 70% recall, TAR 2.0 wins by about 5,000 documents, at 85% recall by less than 1,000 documents, and at 90% recall by over 41,000 documents. Of course, there is
always the question of whether it is possible to hit this curve by training neither too little nor too long. Glossing over that issue for the moment, we assume that we’ve been able to achieve this gain curve by training for exactly the correct amount of time.

So the question is: Even though TAR 1.0 is close to TAR 2.0 in the raw number of reviewed documents, what is the total cost of review? That total cost necessarily includes not only the review work, but the training work as well. And if it costs more to put eyeballs on a training document than on a review document, that must be factored in.

In the actual matter, there was no cost difference between the best reviewers (who were used as proxy for the experts) and the regular reviewers. They were both paid at approximately the same rate. If there were a cost difference, this analysis could be repeated to take that differential into account. But, in general, a subject matter expert (SME) tends to cost much more per hour than a contract reviewer. A rule of thumb in the industry would be about $50 an hour for the contract reviewer and $400 an hour for the SME. For comparison, we are also going to show an analysis using SMEs paid $100 an hour and $200 an hour.

For this analysis, we make the assumption that all reviewers work at a rate of about 50 documents per hour. Thus, a $50-an-hour reviewer costs $1 per document, a $100-an-hour reviewer costs $2 per document, a $200-an-hour reviewer costs $4 per document and a $400-an-hour reviewer costs $8 per document.

Figure 6 shows the results. Along the y-axis is still the cumulative number of responsive documents found, in simulated review order. For the x-axis, however, instead of showing raw document count, it is showing the dollar amount to review each document. With TAR 2.0, all training may be done by $50-an-hour contract reviewers. Thus, every document costs $1 to review. For TAR 1.0, our analysis assumes expert training at anywhere from $50 to $400 an hour, or $1 to $8 per document, and then batched out review at the contract rate of $1 per document.
Figure 6: Based on the gain curves in Figure 5, the monetary cost of using regular reviewers doing TAR 2.0 (blue) versus one of the better expert reviewer TAR 1.0 results (various red). The length of the dotted lines for each TAR 1.0 result, from short to long, indicate an expert reviewer that costs $50, $100, $200 and $400 per hour, respectively.

Under the assumption that the expert costs the same as the contract reviewer, the difference between TAR 1.0 and TAR 2.0 is the same on a cost basis as it is on a total document review count basis. For example, the gap in cost at 85% recall would be less than $1,000. However, if the expert costs $400-an-hour, the cost would be $158,000—over $100,000 more than the TAR 2.0 review, even though the difference in documents reviewed would be less than 1,000.

For additional comparisons, we show two more gain curves and their associated cost curves. Figures 7 and 8 show the gain curve and associated cost curves after training for approximately 25,200 documents. Training takes longer, but in the gain curve the TAR 1.0 approach catches up to the TAR 2.0 approach at about 94% recall. However, the cost to catch up to that gain curve, as shown in Figure 7, is much larger—perhaps even prohibitively so—because of the additional expert training cost.
Figure 7: TAR 2.0 (blue) vs. TAR 1.0 (red) gain curves for 25,200 documents of TAR 1.0 training.

Figure 8: Based on the gain curves in Figure 7, the monetary cost of using regular reviewers doing TAR 2.0 (blue) versus one of the better expert reviewer TAR 1.0 results (various red). The length of the dotted lines for each TAR 1.0 result, from short to long, indicates an expert reviewer that costs $50, $100, $200 and $400 per hour, respectively.

Finally, Figures 9 and 10 show the gain curve and associated cost.
curves after training for approximately 7,200 documents. The training costs are less, but the resulting TAR 1.0 batched-out gain curve is also less effective, which makes the total cost to get to the same level of recall higher as well.

![Figure 9: TAR 2.0 (blue) vs. TAR 1.0 (red) gain curves for 7,200 documents of TAR 1.0 training](image)

Figure 10: Based on the gain curves in Figure 9, the monetary cost of using regular reviewers doing TAR 2.0 (blue) versus one of the better expert reviewer TAR 1.0 results (various red). The length of the dotted lines for each TAR 1.0 result, from short to long, indicates an expert reviewer that costs $50, $100, $200 and $400 per hour, respectively.

![Figure 10: Based on the gain curves in Figure 9, the monetary cost of using regular reviewers doing TAR 2.0 (blue) versus one of the better expert reviewer TAR 1.0 results (various red). The length of the dotted lines for each TAR 1.0 result, from short to long, indicates an expert reviewer that costs $50, $100, $200 and $400 per hour, respectively.](image)
One more figure may be of interest. Figure 11 shows the cost curves of the TAR 1.0 system after training on 7,200, 14,400 and 25,200 documents, all presuming a $200-an-hour expert. From Figure 11, we see that these three TAR 1.0 curves with different amounts of training hit 90% recall at vastly different points, and at vastly different raw numbers of total relevant documents. However, when one takes the cost of reviewing the documents—not just the raw number—each of these techniques hits 90% recall at about the same point, at about $138,000.

What this means is that one can use more expert training and get a better ranking, but the value of a better ranking is offset by the additional cost of the expert to get to that better ranking.

Figure 11: Based on the gain curves in Figures 5, 7 and 9, the monetary cost of using regular reviewers doing TAR 2.0 (blue) versus one of the better expert reviewer TAR 1.0 results (various red). We presume a $200-per-hour expert in all cases.

Furthermore, the TAR 2.0 approach still gets to the same 90% recall point at a cost of about $65,000—a savings of $76,000. We can conclude, therefore, that the TAR 2.0 approach is not only better in terms of raw document counts, but also in terms of total cost savings.

**Expert-Attributable Bottlenecks**

The final analysis that we perform is of elapsed time. One advantage of TAR 2.0 is that one can hit the ground running with one’s entire
review team. By contrast, TAR 1.0 necessitates that the experts finish training before the rest of the review team can start their work. This creates a bottleneck, as there are usually many fewer experts than there are contract reviewers.

We begin with the gain curves from Figure 5. Though the TAR 2.0 approach is ahead at all points, these curves are relatively close at the vast majority of recall points. So, using these curves as the basis, we calculate the time it took to achieve these recall levels.

We know that, in this case, the firm started with eight reviewers and moved to four core reviewers for the later stages of the review. For simplicity's sake, we average that to a review team of six across the entire review. Furthermore, we know that two of those reviewers were the skilled expert reviewers. So we presume a TAR 2.0 review team of six people across the entire review, whereas for the TAR 1.0 workflow, we presume two people doing training and six people doing the batched-out review. Finally, we assume a review rate of about one document per minute. In the TAR 2.0 approach, all six reviewers work in parallel throughout the entire process. In the TAR 1.0 approach, two reviewers work in parallel during training, and six reviewers work in parallel during batched-out review. These assumptions let us create the time-based gain curves in Figure 12.

Figure 12: Time-based gain, assuming different levels of parallelism in the document training and/or review, as per the various workflows:
TAR 2.0 (blue) and TAR 1.0 (red). The x-axis given in 8-hour days (i.e. 15 days equals three 40-hour weeks) and assumes a rate of one minute per reviewer per document.

As you can see, TAR 1.0’s expert-based training is a real bottleneck in the overall elapsed time. Even though the curves in Figure 5 are quite similar to each other, the fact that only two reviewers work in parallel during training almost doubles the amount of time (13.5 days under TAR 2.0, 25 days under TAR 1.0) to get to 80% recall, and it more than doubles the amount of time (22 days under TAR 2.0, 45 days under TAR 1.0) to get to 90% recall.

In fact, after 15 days, the TAR 1.0 system has just barely finished its training, whereas at that same point, the TAR 2.0 system has hit a respectable 84% recall. TAR 1.0 has only finished training when TAR 2.0 has already put eyeballs on just about everything that it needs to.

**Conclusion**

The TAR 2.0 “training and review using everyone” workflow outperforms the TAR 1.0 expert-only training and limited learning workflow, not only in terms of the raw number of documents that need to be reviewed, but also in the cost of doing the review, the time it takes to do the review and the ease with which the review can be done.

---

**Footnotes**

Technology-assisted review (TAR) has a transparency problem. Notwithstanding TAR’s proven savings in both time and review costs, many attorneys hesitate to use it because courts require “transparency” in the TAR process.

Specifically, when courts approve requests to use TAR, they often set the condition that counsel disclose the TAR process they used and which documents they used for training. In some cases, the courts have gone so far as to allow opposing counsel to kibitz during the training process itself.

Attorneys fear that this kind of transparency will force them to reveal work product, thoughts about problem documents, or even case strategy. Although most attorneys accept the requirement to share keyword searches as a condition of using them, disclosing their TAR training documents in conjunction with a production seems a step too far.

Thus, instead of using TAR with its concomitant savings, they stick to keyword searches and linear review. For fear of disclosure, the better technology sits idle and its benefits are lost.
The new generation of TAR engines (TAR 2.0), particularly the continuous active learning (CAL) protocol, however, enable you to avoid the transparency issue altogether.

**Transparency: A TAR 1.0 Problem**

Putting aside the issue of whether attorneys are justified in their reluctance to use TAR, which seems fruitless to debate, consider why this form of transparency is required by the courts.

**Limitations of early TAR.** The simple answer is that the need for transparency arose as an outgrowth of early TAR 1.0 protocols, which used one-time training against a limited set of reference documents (the “control set”). The argument seemed to be that every training call (plus or minus) had a disproportionate impact on the algorithm in that training mistakes could be amplified when the algorithm ran against the full document set. That fear, whether founded or not, led courts to conclude that opposing counsel should be able to oversee the training to ensure costly mistakes weren’t made.

**Solutions in new TAR.** This is not an issue for TAR 2.0, which eschews the one-time training limits of early systems in favor of CAL that continues to train and learn through to the end of the review. This approach minimizes the impact of reviewer “mistakes” because the rankings are based on tens and sometimes hundreds of thousands of judgments, rather than just a few.

CAL also puts to rest the importance of initial training seeds because training continues throughout the process. The training seeds are in fact all of the relevant documents produced at the end of the process. There is no need to scrutinize the initial training documents with CAL, because in stark contrast with TAR 1.0 training, it doesn’t matter which documents you start with. You’ll still wind up with very similar results. In fact we’ve devoted a whole chapter to examining this topic (chapter 26: 57 Ways to Leave Your (Linear) Lover). At most you can debate about the documents that are not produced, whether they have been reviewed or simply discarded as likely non-relevant, but that is a different debate.
This point recently received judicial acknowledgement in an opinion issued by U.S. Magistrate Judge Andrew J. Peck, *Rio Tinto PLC v. Vale S.A.*, 2015 BL 54331 (S.D.N.Y. 2015). Discussing the broader issue of transparency with respect to the training sets used in TAR, Judge Peck observed that CAL minimizes the need for transparency.

*If the TAR methodology uses “continuous active learning” (CAL) (as opposed to simple passive learning (SPL) or simple active learning (SAL), the contents of the seed set is much less significant.*

Let’s see how this works, comparing a typical TAR 1.0 to a TAR 2.0 process.

**TAR 1.0: One-Time Training Against a Control Set**

**A Typical TAR 1.0 Review Process.** As depicted, a TAR 1.0 review is built around the following steps:

1. A subject matter expert (SME), often a senior lawyer, reviews and tags a sample of randomly selected documents (usually about 500) to use as a “control set” for training

2. The SME then begins a training process often starting with a seed set based on hot documents found through keyword searches.

3. The TAR engine uses these judgments to build a classification/ranking algorithm that will find other relevant documents. It tests the algorithm against the 500-document control set to gauge its accuracy.
4. Depending on the testing results, the SME may be asked to do more training to help improve the classification/ranking algorithm. This may be through a review of random or computer-selected documents.

5. This training and testing process continues until the classifier is “stable.” That means its search algorithm is no longer getting better at identifying relevant documents in the control set.

Once the classifier has stabilized, training is complete. At that point your TAR engine has learned as much about the control set as it can. The next step is to turn it loose to rank the larger document population (which can take hours to complete) and then divide the documents into categories to review or set aside depending on their ranking.

**Importance of control set.** In such a process, you can see why emphasis is placed on the development of the 500-document control set and the subsequent training. The control set documents are meant to represent a much larger set of review documents, with every document in the control set standing in for what may be a thousand review documents. If one of the control documents is improperly tagged, the algorithm might suffer as a result.

Of equal importance is how the SME tags the training documents. If the training is based on only a few thousand documents, every decision could have an important impact on the outcome. Tag the documents improperly and you might end up reviewing a lot of highly ranked irrelevant documents or missing a lot of lowly ranked relevant ones.

**TAR 2.0 Continuous Learning; Ranking All Documents Every Time**

TAR 2.0 systems don't use a control set for training. Rather, they rank all of the review documents every time on a continuous basis. Modern TAR engines don't require hours or days to rank a million documents. They can do it in minutes, which is what fuels the continuous learning process.

A TAR 2.0 engine can continuously integrate new judgments by the
review team into the analysis as their work progresses. This allows the review team to do the training rather than depending on an SME for this purpose.

It also means training is based on tens or hundreds of thousands of documents, rather than relying on a few thousand seen by the expert before review begins.

**TAR 2.0: Continuous Active Learning Model**

As the illustration demonstrates, the CAL process is easy to understand and simple to administer. In effect, the reviewers become the trainers and the trainers become reviewers. Training is review, we say. And review is training.

**CAL Steps**

1. Start with as many seed documents as you can find. These are primarily relevant documents which can be used to start training the system. This is an optional step to get the ranking started but is not required. It may involve as few as a single relevant document or many thousands of them.

2. Let the algorithm rank the documents based on your initial seeds.

3. Start reviewing documents based in part on the initial ranking.

4. As the reviewers tag and release their batches, the algorithm continually takes into account their feedback. With increasing
numbers of training seeds, the system gets smarter and feeds more relevant documents to the review team.

5. Ultimately, the team runs out of relevant documents and stops the process. We confirm that most of the relevant documents have been found through a systematic sample of the entire population. This allows us to create a yield curve as well so you can see how effective the system was in ranking the documents.

What Does This Have to Do With Transparency?

In a TAR 2.0 process, transparency is no longer a problem because training is integrated with review. If opposing counsel asks to see the “seed” set, the answer is simple: “You already have it.”

Every document tagged as relevant is a seed in a continuous review process. And every relevant non-privileged document will be or has been produced.

Likewise, there is no basis to look over the expert’s shoulder during the initial training because there is no expert doing the training.

Rather, the review team does the training and continues training until the process is complete. Did they mis-tag the training documents?

Take a look and see for yourself.

This eliminates the concern that you will disclose work product through transparency. With CAL, all of the relevant documents are produced, as they must be with no undue emphasis placed on an initial training set. The documents, both good and bad, are there in the production for opposing counsel to see. But work product or individual judgments by the producing party are hidden. Voila, the transparency problem goes away.

Postscript: What About Other Disclosure Issues?

Discard pile issues. In addressing the transparency concern, we don’t mean to suggest there are no other disclosure issues to be addressed. For example, with any TAR process that uses a review
cutoff, there is always concern about the discard pile. How does one prove that the discard pile can be ignored as not having a significant number of relevant documents?

That is an important issue, one that we discuss elsewhere in this book.

The problem doesn't go away with TAR 2.0 and CAL, but it is the same issue advocates have to address in a TAR 1.0 process or even with a linear review, assuming that culling was used. The answer requires sampling of the unreviewed population.

**Tagging for non-relevance; collection:** What about documents tagged as non-relevant by the review team? How do I know that was done properly?

This too is a separate issue that exists in both TAR 1.0 and 2.0 processes. Indeed, it also exists with linear review. Just as with the discard pile, you must sample the documents coded as non-relevant if you want to assess their accuracy.

And, last but not least, did I properly collect the documents submitted to the TAR process? Good question but, again, a question that applies whether you use TAR or not. Unless you collect the right documents, no process will be reliable.

Again, these are issues to be addressed elsewhere. They exist in whichever TAR process you choose to use. My point here is that with a TAR 2.0 process, a number of the transparency issues that bug people and have hindered the use of this amazing process simply go away.
What Roles Do SMEs and Reviewers Play in a TAR 2.0 World?

With the advent of technology-assisted review (TAR 1.0), the review industry created a new position called a subject matter expert (SME). That person, often a costly senior lawyer and case expert, was tasked with teaching the TAR algorithm how to distinguish between relevant and not. You can read more about this in chapters two and three.

The reason for the allocation of responsibility in a TAR 1.0 protocol is fairly straightforward. TAR 1.0 tools focused on developing models that could be used to classify or rank the collection so that a presumptively relevant or responsive set could be generated for production review. Someone had to review at least several thousand documents to train the algorithm and develop the model. And those training decisions had to be correct, or a model would be inaccurate and the presumptively relevant set would be incomplete.

In a TAR 2.0 world, SMEs get to rejoin the trial team and focus on their case rather than looking at 3,000+ marginally relevant documents for training. Rather, training is done at multiple levels both by the review and the trial teams. The review teams train as they
review, which is the job for which they were hired. The trial team can focus on understanding the case and searching for evidence through other means, including keyword search. As they mark documents relevant, the algorithm learns from them as well. Because TAR 2.0 uses continuous active learning (CAL), as opposed to a one-time training regime, the impact of occasionally inaccurate coding decisions is not as critical. And, senior team members can ensure proper training through algorithmic QC, which we discuss in chapter 16.

Getting rid of the SME and letting the review team handle training was unsettling to many versed in the TAR 1.0 orthodoxy. The fear was that the SME was the real expert and, as such, the only one qualified to judge the documents for proper training. We never believed this was true, arguing that the review team could train as well as an SME but at a lower cost. We also noted that our approach would eliminate one of the biggest negatives with TAR 1.0—namely, the time (weeks sometimes) the review team had to sit on their heels waiting for training to finish so they could begin.

**SME vs. Review Training: Testing the Hypothesis**

We set out to test our hypothesis that reviewers backed by algorithmic QC would train as effectively as an SME. Algorithmic QC is a feature we developed especially for Insight Predict to detect the most likely coding errors (see chapter 16 for an in-depth look). Our plan was to run simulations on the same data for SME versus reviewer training. We also tested review training backed by algorithmic QC to see how it contributed to overall efficiency.

In order to evaluate these different approaches (SME, review only, review plus QC), we used data from the 2009 TREC program. That TREC collection consisted of a large volume of publicly available Enron documents. We primarily used coding judgments on those documents (i.e., relevant to the inquiry or not) that were provided by a team of reviewers hired by TREC for that purpose.

In many cases, we also had judgments on those same documents made by the topic authorities for each of the topics used in our study. The topic authorities were responsible for making final judgments on any disputed coding decisions, and therefore the topic authorities were considered to be SMEs for our research.
We ran simulations on the four TREC topics with available data, and for each topic we evaluated each of the three training methods. For the contract review analysis, we simply used the coding decisions applied by the contract review team. For the SME analysis, we substituted the decision of the SME when the contract review decision was overturned. And for the combined SME QC analysis, we used algorithmic QC to evaluate the top 10% of apparently inaccurate contract reviewer coding decisions, substituting contrary decisions by the topic authority SMEs. To compare their effectiveness, we plotted the yield curves for each review protocol.

The yield curves for the first three TREC topics we considered are shown below. In each case, the lines are almost indistinguishable; the contract review team located the responsive documents just as quickly as the SME. And the QC effort was just as effective up to the 80% recall level.
The results of the analysis of the fourth topic were actually a little more interesting, as can be seen from the yield curve below. For the fourth topic, the relative efficiency of the SME and the contract review team diverged above about 50% recall. This result may be an outlier. But it may also be the result of the typically conservative nature of SMEs compared with contract reviewers, who tend to be liberal in assessing responsiveness. A conservative approach will tend to locate fewer responsive documents, and could therefore inhibit training of the algorithm.
Conclusion

Ultimately, this research supports our belief that it is not necessary to use an SME to train the CAL algorithm in Insight Predict. Contract reviewers are equally effective, if not more so, in locating responsive documents with a TAR 2.0 system utilizing continuous active learning. This allows the trial team (including the SME) to focus on their case and for the SME to focus on QC, to make sure that the review is accurate and consistent (as we address in chapter 16). With contract reviewers taking the laboring oar and an SME steering the ship, the overall review will be as efficient, accurate and cost-effective as possible.

Footnotes

1. In fact, since CAL trains the algorithm until most of the responsive documents have been reviewed and coded, using a single SME to do the training would be impractical. That would essentially mean that a senior attorney would have to review every responsive document being produced.

2. The Text REtrieval Conference is sponsored by the National Institute for Standards and Technology (http://trec.nist.gov/).
An End to Family Batching: Reviewing on a Document Level Is More Efficient

Lawyers have been reviewing document families as a collective unit since well before the advent of technology-assisted review (TAR). They typically look at every document in the family to decide whether the family as a whole (or any part of it) is responsive and needs to be produced, or withheld in its entirety as privileged. Most lawyers believe that is the most efficient way to conduct a review.

What’s the Problem?

There are two problems with reviewing on a family level when you are using a TAR tool.

The first problem is that TAR tools operate at a document level, and only at a document level. So coding decisions have to be made on the text within the four corners of every document. Extraneous text, including the text of attachments or parents, regardless of how overwhelmingly responsive, simply cannot influence the decision on responsiveness for any document under consideration.
Second, reviewing and tagging document families together actually impairs the efficiency of a continuous active learning (CAL) tool. A TAR review will be much more efficient if you batch and review documents individually, i.e., on a document level. In fact, our research has shown that review efficiency is optimized (at least for a CAL review) when documents are batched individually, coded for responsiveness only, and then passed with family members for any further full-family review (assuming production, or privilege withholding, on the basis of entire families).

The “Four Corners”

The most critical point in a TAR review is that coding decisions must be made solely on the basis of the text within the individual document being reviewed. Every coding decision essentially tells the TAR algorithm that the features (text) of the document being reviewed are collectively either responsive or non-responsive. If you make coding decisions on the basis of extraneous text outside the four corners of the document, e.g. on an attachment of the email currently being considered, those decisions may mischaracterize the document and make it more difficult for the TAR algorithm to efficiently rank the remainder of the collection.

The easiest way to implement the four-corners approach from the TAR perspective is to ask this simple question: Do I really want to see more documents that have text like this? If the answer is yes, the document should be coded as responsive, which will inform the TAR tool (at least a CAL tool) to look for more of them. If the answer is no, the document should be coded as non-responsive, so no further time will be wasted on similar documents.

The best example of this four-corners-decision-making process is the review of a parent email that says “Please see attached.” It turns out that the attached is one of the most critical documents in the case. While the attachment is certainly responsive, there is no need to see more documents that simply say “Please see attached.” That language has nothing whatsoever to do with the case, and there is no guarantee that the next attachment will be anything of consequence. Accordingly, the email should be coded on a document level as non-responsive.
There are certainly some instances where referring to family members to determine responsiveness of the underlying document is important, but those are few and far between. Typically, that arises when the text within the four corners of the document is ambiguous, and the family members add context. Building on the previous example, a document that says “Please see attached concerning Project X” is ambiguous on its face. If the attachments suggest that Project X is pertinent to the litigation, then there is indeed a need for the TAR tool to find more documents about Project X. That email would be coded as responsive, even on a document level.

**Family Tagging**

The second problem stems from reviewing on a family level. Intuition should tell us that batching document families together for a TAR review is not the most efficient approach. The reason is that you will be required to review a lot of non-responsive families (and not just individual documents) during the process.

Every TAR ranking will include a percentage of relevant and non-relevant documents. In many cases, as we discuss in another chapter, the ratio is one non-relevant to one relevant. The goal in a TAR review is to review as few non-relevant documents as possible. Even if you choose to review the families of identified responsive documents, that number will be less than if you review the families of documents put forth by the TAR tool that don’t prove to be responsive.

**Simulation Research: Comparing the Approaches**

Our research has confirmed that a document-level Insight Predict review will be more efficient than a family-batched review, even if you eventually review all the family members of the responsive documents. And if you do need to review all of those family members, the best workflow is to first do a document-level Predict review for responsiveness only, and then conduct a comprehensive review of the entire family of every responsive document (i.e., a review for responsiveness, as well as privilege, issue coding, etc.).

To evaluate the benefit of batching and reviewing on a document level, we conducted a simulation of an actual family-batched TAR review. The case had roughly 250,000 documents, and nearly 30,000 of them were responsive (for a richness of about 12%). The initial TAR
review was conducted on a family-batched basis, and we used the final coding judgments to simulate both a family-batched review and a document-level review, from the same starting point (see chapter 21 for an in-depth look at the simulation process).

1. Family-batch review

When we simulated a family-batched review, it was necessary to review about 70,000 total documents to achieve 80% recall. This included every responsive and non-responsive document prioritized for review by Predict, as well as the family members of each of those documents.

2. Document-level review

We then simulated a document-level review to see how many individual documents would have to be reviewed to achieve the same 80% level of recall, before moving on to the family member review. That required the review of only about 36,000 documents, as shown in the below figure.

This figure shows the yield curves for the family batched (red line) and document-level (blue line) reviews. The solid green line reflects the number of documents required to be reviewed to achieve 80% recall on a family-batched basis, and the dashed green line shows the number to be reviewed with a document-level review.
3. Two-phase review

Once we knew how many, and which, documents had to be reviewed to reach 80% recall on a document-level, we expanded the simulation to account for the review of the family members of responsive documents. At that point, we only needed to review an additional 17,000 family members, for a total of roughly 53,000 documents, representing a 24% improvement over a family-batched review.

The chart below shows the previous yield curves for the family-batched and document-level reviews. The review of additional family members is then shown as a branch extending from the blue (document-level) curve. The solid and dashed green lines illustrate the 24% savings when family members of responsive documents are reviewed.²

![Graph showing two-phase review](image)

4. Refined two-phase workflow

We have also determined that even greater levels of efficiency can be achieved by following a refined two-phase workflow. The first phase would be a document-level Predict review solely for the purpose of identifying responsive documents. When a CAL algorithm is continuously ranking the entire collection on the basis of relevance or responsiveness, a responsiveness-only review can proceed much more quickly than a comprehensive document review.³

Responsive documents would then be passed, together with all family members, to a comprehensive family review for production.
Family review can be either simultaneous or sequential, i.e., at the end of the responsiveness review.

To evaluate the effectiveness of this workflow, we ran post-hoc simulations on several different collections, each of which had been fully coded for production. These simulations were slightly different, however, because we had to estimate the time it would take to review documents, in addition to the order in which they were reviewed.

As a baseline, we simulated a typical document-level Predict review, followed by comprehensive review of any remaining family members. In a typical review, every document is reviewed only once, for every coding issue—responsiveness, privilege, etc. So every review decision is estimated to take the same average time, regardless of whether it was made during the Predict review or the family review.

For comparison, we simulated the refined two-phase workflow—an expedited first-phase Predict responsiveness-only review, followed by a comprehensive review of the entire responsive family (including the document coded in the first phase). Since the family review is essentially the same type of review as the baseline, we estimated the family review at the same average review time as the baseline review. And, since we have seen responsiveness-only reviews proceed at a much faster pace than a comprehensive review, we evaluated different Predict review rates of one- to five-times the rate of a comprehensive review.

The chart on the following page shows the time-based gain curves for just one of the collections that we evaluated. On the x-axis, we plotted the total time it would take to review sufficient documents to achieve the recall levels that are plotted on the y-axis. The thick, dark red line is the baseline, which assumes the same average review rate for every document. The other red lines represent the refined workflow at five different Predict responsiveness-only review rates, which are indicated for each curve.

As this chart shows, the refined two-phase workflow is as efficient, or more efficient, than a typical document-level Predict review when the responsiveness-only review is at least twice as fast as the average for the comprehensive review. And the results of the other simulations were very consistent.
Conclusion

Ultimately, our simulations show that reviewing on a document level will be more efficient than family batching. And by refining the workflow to focus first on responsiveness alone, review rates and efficiency can improve even further.

Footnotes

1. In reality, the likelihood that a TAR tool would elevate that particular email as a potentially responsive document is low, for exactly the reason it must be coded as non-responsive—the text within the four corners of the document simply has no bearing on the case. (Conversely, the attachment would likely be elevated for review much earlier in the TAR process.)

2. It should be noted that reviewing the family members actually increased recall to 85%.

Using Algorithmic QC to Correct Review Mistakes

Quality control (QC) is an essential part of every review, with the only questions being: who, how deep (what percentage) and how often. The typical practice is to take a random sample of a reviewer’s output and measure it against the “gold standard,” which would be how a more senior reviewer might tag the same record. The number of overturns becomes a measure of the reviewer’s reliability and a statement about the quality of the linear review.

We add a second layer of QC through an algorithm we developed especially for Insight Predict. We call it algorithmic QC and it can identify coding decisions that are not only outliers, but appear to be true errors. By using algorithmic QC early and periodically throughout the review process, these coding errors can be located and corrected to ensure the consistency and accuracy of the review.

How Algorithmic QC Works

The typical approach to quality control of a TAR ranking is what can be termed outlier identification. That means looking for the negatively-coded documents at the top of the ranking where the most-likely responsive documents are located, and looking for the
positively-coded documents at the bottom of the ranking, where the documents are primarily negative. In the figure below, the highlighted positions represent the outliers that would typically be identified for quality control in a TAR review. The inherent deficiency in this approach is that, because it is based on the ranking of the entire collection, it uses any improperly coded documents to essentially rank themselves.

The Predict algorithmic QC protocol works differently. It does not look at the entire ranking to locate the outliers. Instead, algorithmic QC focuses separately on the responsive (positive) documents and the non-responsive (negative) documents.

Each time Predict re-ranks the entire collection, algorithmic QC generates separate, independent models of both the positive and negative documents. Then it ranks the positive documents by how much they look like the negative model, and it ranks the negative documents by how much they look like the positive model. In other words, the documents at the top of the positive ranking look like they are negative, and vice versa. The figure below shows the structure of the two rankings.
Using these two rankings, documents batched for QC review are selected from the top of each ranking. That way, algorithmic QC looks for coding errors, not just outliers. And with two separate rankings, the QC review process can focus on either the positive documents, the negative documents, or both.

**Optimizing the QC Process**

Quality control is just that—a qualitative process. There are no precise quantitative guidelines for managing a Predict review, and every review will be different. However, using algorithmic QC early in the review process, and periodically throughout the process, will certainly enhance the consistency and accuracy of the review.

Algorithmic QC should be initiated early in the Predict review process to make sure the review team is on the right track, both individually and collectively. For algorithmic QC to be fully informative, there should be positive and negative coding decisions by every reviewer on the team, so the perspectives and decisions of every reviewer can be evaluated. There is no particular timing, however, since algorithmic QC is always operating in the background. It simply needs to be early enough to provide a check on the relevance judgments of the review team, without having to spend an inordinate amount of time rectifying any errors that might be found.

After the first round, algorithmic QC should be repeated periodically throughout the review process. Again, there is no particular timing, but one approach is to conduct algorithmic QC every time the estimated recall increases by increments of 10% or 20%. In addition, the algorithmic QC process should be used any time the scope of responsiveness changes, in order to quickly confirm previous decisions.

In order to ensure the consistency and accuracy of the coding decisions made throughout the review, a SME is typically responsible for the algorithmic QC process. And there are really two goals to the QC process. Since every coding decision trains the Predict algorithm, locating and rectifying coding errors will improve the algorithm, and make the review more efficient. At the same time, algorithmic QC will illuminate any misperceptions on the part of the review team as to the scope of responsiveness, so they can be promptly corrected.
The size of algorithmic QC batches should be consistent, typically 50 or 100 documents, to make it easier to determine when each round can be considered complete. There will always be some level of disagreement among reviewers, so a SME should focus on obvious errors in judgment rather than close calls that are really just a matter of degree. Successive QC batches should be reviewed until those errors in judgment have been identified, and the level of disagreement between the reviewers and the SME in QC batches has diminished.¹

Footnotes

1. Significant, consistent coding errors that are identified in the algorithmic QC process can often be rectified en masse using advanced analytics.
A critical metric in technology-assisted review (TAR) is recall, which is the percentage of relevant documents actually found in the collection. One of the most compelling reasons for using TAR is the promise that a review team can achieve a desired level of recall (say 75%) after reviewing only a small portion of the total document population (say 5%). The savings come from not having to review the remaining 95% of the documents. The argument is that the remaining documents (the “discard pile”) include so few that are relevant (among so many non-relevant documents that are left) that further review is not economically justified.

How do we prove we have found a given percentage of the relevant documents at whatever point we stop the review? Some suggest you can prove recall by sampling only a relatively few documents, which is not statistically valid. Others suggest approaches that are more statistically valid, but require sampling a lot of documents (as many as 34,000 in one case). Either way, this presents a problem. Legal professionals need a reasonable but also statistically reliable way to measure recall in order to justify review cutoff decisions.
A Hypothetical Review

To illustrate the problem, let’s conjure up a hypothetical review. Assume we collected one million documents. Assume also that the percentage of relevant documents in the collection is 1%. That suggests there are 10,000 relevant documents in our collection (1,000,000*.01).

Using Sampling to Estimate Richness

Typically we don’t know in advance how many relevant documents are in the collection. To find this information, we need to estimate the collection's richness (aka prevalence) using statistical sampling, which is simply a method in which a sample of the document population is drawn at random, such that statistical properties of the sample may be extrapolated to the entire document population.

To create our sample we must randomly select a subset of the population and use the results to estimate the characteristics of the larger population. The degree of certainty around our estimate is a function of the number of documents we sample.

While this is not meant to be a chapter about statistical sampling, here are a few concepts you should know. Although there are many reference sources for these terms, we will draw from from the excellent *The Grossman-Cormack Glossary of Technology-Assisted Review*, 7 Fed. Cts. L. Rev. 1 (2013):

1. **Point estimate**: The most likely value for a population characteristic. Thus, when we estimate that a document population contains 10,000 relevant documents, we are offering a point estimate.

2. **Confidence interval**: A range of values around our point estimate that we believe contains the true value of the number being estimated. For example, if the confidence interval for our point estimate ranges from 8,000 to 12,000, that means we believe the true value will appear within that range.

3. **Margin of error**: The maximum amount by which a point estimate might deviate from the true value, typically expressed as
a percentage. People often talk about a 5% margin of error, which simply means the expected confidence interval is 5% above or below the point estimate.

4. **Confidence level:** The chance that our confidence interval will include the true value. For example, “95% confidence” means that if one were to draw 100 independent random samples of the same size, and compute the point estimate and confidence interval from each sample, about 95 of the 100 confidence intervals would contain the true value.

5. **Sample size:** The number of documents we have to sample in order to achieve a specific confidence interval and confidence level. In general, the higher the confidence level, the more documents we have to review. Likewise, if we want a narrower confidence interval, we will have to increase our sample size.

It might help to see these concepts displayed visually. Here is a chart showing what a 95% confidence level looks like against a “normal” distribution of document values as well as a specific confidence interval.

![Confidence Interval Chart](image)

**Point Estimate and Confidence Interval**

In this case, our point estimate was 500 relevant documents in our collection. Our confidence interval (shaded) suggests that the actual range of relevant documents could go from 460 at the lower end of our estimate to 540 at the higher end.

Part of the curve is not shaded. It covers the 5% chance that the actual number of relevant documents is either above (2.5%) or below (2.5%) our confidence interval range.

**Our Hypothetical Estimate**

We start our analysis with a sample of 600 documents, chosen randomly from the larger population. The sample size was based on
a desired confidence level of 95% and a desired margin of error of 4%. You can use other numbers for this part of the exercise but these will do for our calculations.

How did we get 600? There are a number of online calculators you can use to determine sample size based on your choices about confidence levels and margin of error. We recommend the Raosoft calculator because it is free and simple to use.

As you can see, we entered the population size (1,000,000), a desired confidence level (95%), and a margin of error (4%). In turn, the calculator suggested that we look at 600 documents for our sample.

**Initial Sampling Results**

Let's assume we found six relevant documents out of the 600 we sampled. That translates to 0.01 or 1% richness (6/600). We can use that percentage to estimate that there are 10,000 relevant documents in the total review population (1,000,000*.01). This becomes our point estimate.

What about the margin of error? In this case we chose a sample size that would give us up to a 4% margin of error. That means the estimated number of relevant documents in our population is within a 4% range +/- of our point estimate of 10,000 documents.

As noted, there are a million documents in the collection; 4% of 1 million comes to 40,000 documents. If we use that figure for our margin of error, it suggests that our confidence interval for relevant documents could range from the six we found in our sample to as high as 50,000. That is an interesting spread.

**Determining the Exact Confidence Interval**

In practice we would use a more refined approach to calculate our
confidence interval. It turns out that the “exact” confidence interval depends on the results of the random sample. In this case we will use a binomial calculator to incorporate the survey results to determine our exact confidence interval.

Based on our planned sample size (600) and the number of relevant documents we found (6), our confidence interval (expressed as a decimal) ranges from 0.0037 (lower) to 0.0216 (upper). We multiply these decimal values against the total number of documents in our collection (1,000,000) to calculate our exact confidence interval. In this case, it runs from 3,700 to 21,600.

So, we have a start on the problem. We believe there are 10,000 relevant documents in our collection (our point estimate) but it could be as high as 21,600 (or as low as 3,700). Let’s move on to our review.

**The Review**

The team finds 7,500 relevant documents after looking at the first 50,000. Based on our initial point estimate of 10,000 relevant documents, we could reasonably conclude we have found 75% of the relevant material. At that point, we might decide to shut down the review. Most courts would view stopping at 75% recall to be more than reasonable.

Your argument to the court seems compelling. If there were only 2,500 relevant documents left in the discard pile, the cost of reviewing another 950,000 documents to find 2,500 relevant ones seems disproportionate. On average, you would have to look at 380 documents to find the next relevant document. At a cost of $2 per document for review, it would cost $760 for each additional relevant document found. If you continued until the end, the cost would be an extra $1.9 million.
How Do We Know We Achieved 75% Recall?

Now comes the hard part. How do we know we actually found 75% of the relevant documents?

Remember that our initial point estimate was 10,000 documents, which seems to support this position. However, it had a confidence interval which suggested the real number of relevant documents could be as high as 21,600.

That means our recall estimate could be off by quite a bit. Here are the numbers for this simple mathematical exercise:

• We found 7,500 documents during the review.

• If there are only 10,000 relevant documents in the total population, it is easy to conclude we achieved 75% recall (7,500/10,000).

• However, if there were 21,600 relevant documents in the population (the upper range for the confidence interval), we achieved only 35% recall of relevant documents (7,500/21,600).

Those numbers would give grist for an argument that the producing party did not meet its burden to find a reasonable number of relevant documents. While the team may have found and reviewed 75% of the relevant documents, it is also possible that they found and reviewed only 35%. Most would agree that 35% recall is not enough to meet your duty as a producing party.

Sampling the Discard Pile

So what do we do about this problem? One answer is to sample the discard population to determine its richness (some call this term elusion). If we could show that there were only a limited number of relevant documents in the discard pile, that would help establish our bona fides.

Let's make some further assumptions. We sample the discard pile (950,000 documents), again reviewing 600 documents based on our choice of a 95% confidence level and a 4% nominal confidence interval.
This time we find two relevant documents, which suggests that the number of relevant documents in the discard pile has dropped to about 0.33% (2/600). From there we can estimate that we would find only 3,135 relevant documents in the discard pile (950,000*0.0033). Added to the 7,500 documents we found in review, that makes a total of 10,635 relevant documents in the collection.

Using that figure we calculate that the review team found about 71% of the relevant documents (7,500/10,635). While not quite 75%, this is still a number that most courts have accepted as reasonable and proportionate.

**What About the Confidence Interval?**

But how big is our exact confidence interval? Using our binomial calculator, we get this range:

![Binomial Confidence Intervals](image)

Applying these figures to our discard pile, we estimate that there could be as many as 11,400 relevant documents left (0.0120*950,000).

If we add the 7,500 documents already found to the upper value of 11,400 documents from our sample, we get a much lower estimate of recall. Specifically, we are producing 7,500 out of what could be as many as 18,900 relevant documents. That comes to a recall rate of 40% (7,500/18,900).
Is that enough? Again, I suspect most readers—and courts—would say no. Producing just two out of five relevant documents in a population would not seem reasonable.

**Increasing the Sample Size**

What to do? One option is to try to narrow the margin of error (and ultimately the exact confidence interval) with a larger sample. We will narrow the margin of error to 1% and see how that impacts our analysis.

Our calculator suggests we would have to sample 9,508 documents. Assume we find 31 relevant documents out of the 9,508 documents we sampled, which would again support our richness estimate of about 0.33% (31/9,508).

We will enter the sampled richness into our binomial calculator to find out our exact confidence interval.

![Binomial Confidence Intervals]

Applying the confidence interval figures to our discard pile we reach the following conclusions:

1. We estimate there are 3,097 relevant documents in the discard pile, about the same as before (950,000*(31/9508)).

2. The lower range of relevant documents is 2,090 (0.0022*950,000).

3. The upper range of relevant documents is 4,370 (0.0046*950,000).

Using these values for our exact confidence interval, the range goes from 63% (7,500/11,870) to 78% (7,500/9,590). We think most would agree that this type of confidence interval would be reasonable. It
would suggest that you found 70% of the relevant documents in your review, with the understanding that the number might be as low as 63% but could be as high as 78%.

The Cost of Proving Recall

We have found a method to prove recall by sampling the discard pile. But at what cost? If we are satisfied with a recall rate of 54% for the lower boundary of our confidence interval, we would have to sample 2,395 documents. At 100 documents an hour, the sample would take about 24 hours of review to complete. At $2 per document, the cost would be $4,790.

If we decide to narrow the interval and reach a minimum recall rate of 63%, then the sample size quadruples to 9,508 documents. If we again assume 100 documents an hour, review time would go up to 95 hours, which is more than two weeks of effort. At $2 per document, the cost would jump to $19,016.

To make matters worse, what happens if our confirming sample doesn’t support our initial estimate? At that point we would have to continue our review until we found a reasonable percentage. Then we would have to review another sample from the discard pile to confirm that we had indeed found 75% of the relevant documents (or whatever number we end up at).

You now see the problem inherent in proving recall, especially for collections with low richness. It can require a larger sample size than you might otherwise like.
How Many Documents Will I Have to Review?

In the last chapter we talked about measuring recall, which is the percentage of the relevant documents found in the review process. Recall is important because lawyers have a duty to take reasonable (and proportionate) steps to produce responsive documents. Indeed, Rule 26(g) of the Federal Rules effectively requires that an attorney certify, after reasonable inquiry, that discovery responses and any associated production are reasonable and proportionate under the totality of the circumstances.

In that regard, achieving a recall rate of less than 50% does not seem reasonable, nor is it often likely to be proportionate. Current technology-assisted review (TAR) decisions suggest that reaching 75% recall is likely reasonable, especially given the potential cost to find additional relevant documents. Higher recall rates, 80% or higher, would seem reasonable in almost every case.

But recall is only half of the story. Achieving any level of recall comes at a price. That price can be expressed in terms of “precision,” which is the ratio of relevant to non-relevant documents that must be reviewed to reach any level of recall. The cost of review is a function of the precision of your TAR process, just as it is driven by the level of recall you have to attain.
How many documents must be reviewed to find one relevant document in a TAR process? There is no one answer to that question. The precision of any TAR process will depend on a number of factors including the nature of the documents themselves, the algorithm used, the effectiveness of the training process and the level of recall obtained at the point of measurement.

For example, in several studies Maura Grossman and Gordon Cormack have suggested you should expect to review two documents for every relevant one found (based on achieving 75% recall). This amounts to a 50% precision rate which seems pretty good, particularly in collections with low richness.

In an effort to contribute to this discussion, we took a look at three simulations and a dozen cases where our clients used Predict, Catalyst’s advanced TAR 2.0 engine, for their review. Our purpose in doing so was to calculate the precision rates obtained to see if we could discern a pattern. In doing so we recognized that our small sample wasn’t statistically predictive, either in the way the cases were selected or their number. Rather, we had data for these cases and decided we would report on them for whatever value could be derived. At some future point, we hope to aggregate more data on a larger case population and repeat the experiment.

The Projects

We can’t say too much about the cases or the simulations because of client confidentiality. We have named them Sim 1 through Sim 3, and Case 1 through Case 12 as a result. For each we have listed the number of documents in the collection as well as the estimated richness of the collection. And, because the question from clients is most often framed in terms of the number of documents that will need to be reviewed, we essentially show precision as its reciprocal, i.e., the number of documents the team reviewed to find each relevant document.

We named this statistic Predict Efficiency, and we generally want the figure to be as close to 1.0 (a “perfect” review) as possible. Thus, for example, the richness in Case 1 was well under 1%. The team had to review 5.77 documents for each relevant document found. At the other end of the spectrum, in Sim 2, with richness running at almost 42%, the team had to review just one and one-half documents for
each relevant one found. Sim 2 was obviously more efficient, since the team needed to review fewer non-responsive documents to find one responsive document.

<table>
<thead>
<tr>
<th>Project</th>
<th>Docs</th>
<th>Richness</th>
<th>Predict Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sim 1</td>
<td>3,619</td>
<td>19.09%</td>
<td>2.22</td>
</tr>
<tr>
<td>Sim 2</td>
<td>521,669</td>
<td>41.94%</td>
<td>1.50</td>
</tr>
<tr>
<td>Sim 3</td>
<td>243,201</td>
<td>11.88%</td>
<td>1.56</td>
</tr>
<tr>
<td>Case 1</td>
<td>192,019</td>
<td>0.46%</td>
<td>5.77</td>
</tr>
<tr>
<td>Case 2</td>
<td>235,370</td>
<td>24.11%</td>
<td>1.88</td>
</tr>
<tr>
<td>Case 3</td>
<td>24,236</td>
<td>24.86%</td>
<td>2.71</td>
</tr>
<tr>
<td>Case 4</td>
<td>42,432</td>
<td>19.33%</td>
<td>3.08</td>
</tr>
<tr>
<td>Case 5</td>
<td>52,682</td>
<td>10.10%</td>
<td>1.83</td>
</tr>
<tr>
<td>Case 6</td>
<td>105,690</td>
<td>7.05%</td>
<td>3.14</td>
</tr>
<tr>
<td>Case 7</td>
<td>28,856</td>
<td>6.08%</td>
<td>1.81</td>
</tr>
<tr>
<td>Case 8</td>
<td>273,854</td>
<td>1.56%</td>
<td>5.29</td>
</tr>
<tr>
<td>Case 9</td>
<td>131,654</td>
<td>26.02%</td>
<td>2.15</td>
</tr>
<tr>
<td>Case 10</td>
<td>147,552</td>
<td>32.13%</td>
<td>1.92</td>
</tr>
<tr>
<td>Case 11</td>
<td>48,936</td>
<td>20.94%</td>
<td>1.83</td>
</tr>
<tr>
<td>Case 12</td>
<td>935,931</td>
<td>0.98%</td>
<td>3.17</td>
</tr>
</tbody>
</table>

In this regard, we should note that none of these reviews stopped at 75% recall. All went above 80% and many went above 90%. As a result, we would expect to see lower precision figures than might be expected at a 75% recall rate.

Here is a plot showing the precision numbers (as Predict Efficiency) for all 15 projects.
Many of the cases came in at a roughly two to one precision ratio. This is in line with the Grossman Cormack results and, to our knowledge, beats a keyword review by a wide margin.

A few of the cases had higher numbers, e.g. over five to one, but there were typically reasons for this. For example, it is not at all unusual to see only modest precision results with extremely sparse collections like Cases 1, 8 and 12, where richness was fairly low. In several of the other cases, the review continued well beyond the 75% and 80% recall marks, at which point there are simply fewer responsive documents in the collection and, therefore, in the final batches being reviewed.

The average across these cases was just under three to one, reflecting a precision rate of just about 37.5%. Is that a good result for the team? When compared to linear review, there is no question about it. In each case the team was able to achieve a higher than required recall while still reviewing only a fraction of the total population. When compared to linear review, the results are substantially better. With keyword search, we doubt many would achieve similar levels of recall without having to review a far larger percentage of the documents.

We can make one other observation, particularly by comparing the three simulations to the actual case reviews. Each of the simulated reviews was below the average in terms of the number of documents that had to be reviewed to find one responsive document, and two of the three exhibited the best results of all projects.

Ultimately, that is probably not surprising. Real reviews suffer, for example, from incorrect or inconsistent coding that gets fixed through the QC process. What that means practically is that the algorithm is improperly trained at times during the review process, and then rectified through QC. A simulation uses the final coding decisions once a review is finished, so every coding decision that informs the algorithm is correct, and the algorithm is optimally trained. Again, this is not a statistical analysis, but does provide some insight into what you might expect when you are running a Predict review.
Conclusion

As we mentioned at the outset, this is not by any means a statistical analysis of Predict efficiency. But we can observe a few trends, even among these few examples. First, the realities of an actual Predict review—things like coding errors and inconsistency, quality control measures, relevance drift, etc.—will likely make the review less efficient than a perfect, simulated review where the true coding decisions are always known and applied. As a corollary, a thorough, careful and considered review, with less discrepancies, will likely improve efficiency. Second, sparse collections will also likely be less efficient than collections with a more reasonable richness level. Finally, these numbers are not inconsistent with our general observations of the performance of Predict, even outside these specific cases. So, in the absence of a more precise statistical evaluation, you may be able to use this data as a quick rule of thumb to guide your projections as you plan a review directed toward achieving a high level of recall using Predict.
With CAL How Do I Know When to Stop Review? Chapter 19

An important question for technology-assisted review (TAR) professionals is: How do I know when to stop the review with a continuous active learning (CAL) protocol? In chapter 17, we talked about measuring recall, which is an important indicator of whether the review was adequate. In the next chapter we will talk about validation, which is the effort to prove that you achieved adequate recall. This chapter covers the middle portion, focusing on how you make the determination to stop the review and begin the validation process.

**Estimating Review Size**

As a starting point, many review managers would like some idea as to how many documents will have to be reviewed. While there is no guaranteed way to answer that question, at least in advance, you can reasonably estimate in advance the effort that may be required to review a collection to a reasonable degree of completion.

In order to plan your resources, you need to know roughly how many positive documents exist in the collection. That information is best obtained by evaluating a statistical sample of the collection, then extrapolating those results back to the entire collection. With an
estimate of the number of responsive documents in the collection, and an idea of average family size, general CAL studies can give you a rule-of-thumb estimate of when you will be able to stop reviewing and how much effort that will take.

Let us offer an example. Assume you have a collection of 400,000 documents about which you know very little. Take a random sample of 384 documents and code the sample for responsiveness. That will give you a statistical confidence level of 95%, with a maximum 5% margin of error. (Use an online sample size calculator, such as raosoft.com, to determine the sample size you need for a given confidence level.)

If 70 documents are responsive, the richness of the sample is 18.2% (70/384). Since the sample is representative of the entire collection, you can reasonably expect that the collection will also be 18.2% rich—meaning there are roughly 72,800 responsive documents in the collection (0.182 * 400,000). If you are planning to achieve 90% recall, you can expect to stop reviewing once you have found around 65,500 responsive documents (72,800 * 0.9).

After you know roughly how many responsive documents you are looking to find, you can estimate the total number of documents you may need to code to complete your review. As a general rule of thumb, studies by Maura Grossman and Gordon Cormack have suggested that you need to review about twice the number of responsive documents you are seeking in order to achieve a reasonable recall (75% in their studies). In our example, that means you would have to review and code 131,000 documents (~33% of the collection) to locate enough responsive documents to achieve the desired 90% recall.

In most cases, you will also want to review the family members of the responsive documents, for privilege if nothing else. Assume the average family size is 1.3 members. That means you still need to review roughly 20,000 members of the responsive document families (0.3 * 65,500). The sum gives you a decent estimate of the total number of documents you will need to review to achieve 90% recall—about 151,000 documents (131,000 + 20,000).
You Have Started Your Review. How Do You Know When to Quit?

How do you know when you can quit reviewing with CAL? The estimate of the number of responsive documents is certainly one target to keep your eye on during a CAL review. In our example, you can reasonably expect to review at least until you have achieved your objective of finding 65,500 responsive documents.

But the primary indicator for a CAL review is another metric we call batch richness, or batch precision. Batch richness is the fraction of documents that reviewers are coding as responsive during any given period of time. We like to monitor batch richness on a daily basis to make sure reviewers are not wasting time and money by reviewing too many non-responsive documents.

With Insight Predict, you will generally see the batch richness climb quickly to some multiple of the richness of the overall collection. In one recent case, for example, collection richness was 18.5%, and the daily batch richness immediately jumped to nearly 70%. That means that 70% of the documents seen by reviewers were responsive. Not every collection, however, rises to that level. The Grossman-Cormack studies discussed above suggest that the average batch richness for a CAL review should be somewhere around 50%.

Here is an example showing batch richness over the course of an actual review, taken from a case study we describe later in this book:
In this chart, the x-axis represents the number of batches delivered to the reviewers. The y-axis records the percentage of relevant documents seen in the batched (as marked by the reviewer). The red line shows average batch richness over time.

There were almost 5,000 batches in this review. Each blue line represented the percentage of relevant documents in the batch. In the early stages, the number reached 80% to 90%. That meant the reviewer found 80 to 90 responsive documents out of 100 in their batch. It also meant the reviewer saw only a few non-responsive documents, which is the ultimate goal.

In a typical Predict review, batch richness will plateau for some period of time, and then dramatically decline and stabilize at a much lower level. As a rule of thumb, once the batch richness drops to about 10% of the high plateau value, you have likely depleted the pool of responsive documents and achieved a respectable recall. So, for example, if the daily batch richness plateaued at 50% (the Grossman-Cormack average), you can probably quit reviewing once you see daily batch richness decline to about 5% for a few rounds. Using this technique, we are typically achieving recall rates in excess of 90%.

**Taking a Systematic Random Sample**

At a point when you think you have or are close to reaching your goal, there is a simple way to confirm your intuition. With Predict, we recommend taking a specialized type of random sample called a systematic random sample, which we discussed in chapter 3.

In brief, a systematic random sample takes advantage of Predict's ability to rank all documents in your collection (rather than using a small control set). Specifically, Predict chooses documents based on their ranked order, sampling every nth document from the top to the bottom of the ranking (e.g. every 100th document). Using this method helps ensure that we are looking at documents across the ranking spectrum, from highest to lowest. It also helps us draw a yield curve, which we discussed in chapter 3.

Your job is to review the sample (not in any identifiable order) and mark the documents as relevant or non-relevant. Once that is done, Predict will create a yield curve that looks something like this:
With CAL How Do I Know When to Stop Review? Chapter 19

Reaching back to our discussion on yield curves, the x-axis shows the percentage of documents reviewed. The y-axis measures the percentage of relevant documents found during the review. The red line shows the results of a linear (essentially a random) review. The blue line traces the results from your systematic random sample (with it moving up for each relevant document found) and to the right for each non-relevant document found.

Most yield curves have what many call an elbow or a knee. The example above shows a sharp bend at about the 90% point. That tells you that batch richness is likely to drop significantly as you pass the bend. To the left of the bend, there is a relatively high percentage of relevant documents being found. To the right, you have to look at increasing numbers of documents before you find the next relevant one.

We show the numerical values on the right side the report. For this review we reported as follows:

**Left (or Right) of Cutoff**

- Number of documents
- # of reviewed docs
- # of unreviewed docs
- Estimated relevant
- Precision
- Recall
- # docs to next relevant
- Cost per relevant
- Total review cost
- Total review time
These figures are generated using simple calculations based on the sample you created and coded. The cost figures are added by the user and are specific to each matter, and the displayed cost estimates are derived by rather simple math.

The systematic random sample provides an easy way to estimate both your current recall level and also how many documents you would likely need to review to reach any desired recall goal. If the results indicate that you haven’t quite met your recall goal and the cost to find a good number of remaining responsive documents isn’t too high, you might make the decision to keep reviewing for a couple more days. You can always take another systematic random sample at a later point to get updated metrics. Once you reach your goal, it is time to stop and begin the validation process, which we will discuss in the next chapter.
How Do I Validate My Results?

In the preceding chapters, we have spoken about statistics, recall and other measurements that help move the TAR process forward. In this chapter we focus on the tail end of the process. To defend our TAR review decisions, and one could argue any review decisions, we need to validate our results. How do we do that?

Validation is the “act of confirming that a process has achieved its intended purpose.”¹ It is important to TAR for several reasons, including the need to ensure the TAR algorithm has worked properly and because Rule 26(g) requires counsel to certify that the process they used for producing discovery documents was reasonable and reasonably effective.² While courts have approved validation methods in specific cases,³ no court has yet purported to set forth specific validation standards applicable to all cases or for all TAR review projects.

Every validation process involves some form of sampling, either judgmental or statistical.⁴ A judgmental sample is based primarily on subjective choices—for example a keyword search looking for privileged documents missed in review.⁵ In contrast, statistical sampling requires that the sample be drawn randomly from the
entire document population. The key benefit of a statistical sample is that it provides a defensible basis to extrapolate sample results to a larger document population.

**Validating a Recall Estimate**

The goal of TAR is to reduce the number of documents necessary for review. This is typically done through a review “cutoff,” meaning that you stop before all documents are reviewed. In most cases, validation is required to demonstrate that the cutoff point is reasonable. For example, you may want to show that the TAR process led to review and production of 75% of the relevant documents.

The proposition to be validated is that only 25% of the relevant documents were left in the un-reviewed population, often called the “null set” or sometimes the “discard pile.” To validate the proposition you have to estimate recall (which is the percentage of relevant documents found during your review) in a statistically sound manner.

**The Direct Method**

The simplest and arguably the most statistically sound method for estimating recall is what many call the Direct Method. This approach involves a random sample drawn from the entire document population but requires that you continue to sample until you find a sufficient number of relevant documents to meet a required sample size.

Using a freely available sampling calculator, we might determine that we need a sample size of 384 relevant documents to achieve a 95% confidence level and a 5% margin of error.

Once the sample has been taken (and we have found 384 relevant documents),
we can compare the number of documents that were included in the already reviewed production set with the total number in the sample. If, for example, we determined that 288 were part of the reviewed/production population, we would conclude the TAR process found 75% of the relevant documents (288/384), for a point estimate of 75% recall.

Because a point estimate tells only part of the statistical story, we would also need to determine the confidence interval around our estimate. For this purpose we would use what statisticians call a “binomial proportion confidence interval” calculator. Using such a calculator, as you can see below, we would find that the TAR process had promoted from 70% to 79% of the relevant documents for review and production, leaving between 21% and 30% of the relevant documents in the null set.

![Binomial Confidence Intervals](image)

We could use our statistical evidence to validate a claim that we achieved about 75% recall and lay the groundwork for an argument that it was reasonable to stop the review at this point.

**Problems with the Direct Method**

The problem with the Direct Method is the requirement that the sample be composed solely of relevant documents. If the document population is 1% rich, you will need to review 100 documents on average for each relevant one found. In order to find 384 relevant documents in such a case, you might need to sample as many as 38,400 documents. If richness were even lower, say 0.1%, you would have to look at 384,000 documents, on average, in order to obtain
a valid sample. As many have noted, that is a huge and arguably unreasonable burden simply to validate review results.18

Other Approaches to Estimating Recall

There are several other approaches to estimating recall which have been proposed or used in practice. These have been labeled the “ratio methods”19 because they either compare a known value with an estimate or compare two estimates for the recall calculation. For example:

1. The number of relevant documents found during review with the estimated richness of the collection.20
2. The number of relevant documents found during review with the estimated number of documents in the null set.21
3. Estimated richness with the estimated number of relevant documents left in the null set.22

The ratio methods seem logical but can be statistically challenged if the proponent fails to take into account the confidence interval inherent in each point estimate.23 Imagine a scenario where we found 75,000 relevant documents during the review but ended the review with two million documents left in the null set. If we took a sample of the null set for relevant documents, we might estimate that it contained only 25,000 relevant documents (a point estimate), or about 1.25% of the total. This would support an argument that we found and produced 75% of the relevant documents during the review (75,000/100,000), which at least some courts have deemed reasonable.24

If, however, we take the associated confidence interval for our point estimate into account, the recall estimate could be quite different. Assume, for simplicity’s sake, that the upper bound of the confidence interval was 5% above our point estimate of 1.25%. That would suggest that the total number of relevant documents in the unreviewed population could be as high as 6.25% which is the upper bound of the estimate. In terms of documents, that suggests that the number of relevant documents in the null set could be as high as 125,000 documents. In such a case, the review might only have found 37.5% of the relevant documents (75,000/200,000). That number may not be sufficient to meet the reasonableness obligations under Rule 26(g).
Validation Goals

Ultimately, the goal of a TAR validation process is to confirm that you achieved a certain result. In that regard, the burden in validating a TAR review should be no different than that faced by counsel supervising a collection or doing a linear review based on keyword searches.\(^\text{25}\) Whichever approach is taken, Rule 26(g) requires that counsel follow a reasonable process to identify relevant documents and validate the results (including those not reviewed) in some statistically sound fashion. By practical necessity, the methodology chosen will require you to exercise judgment and balance the effort required against the benefits achieved through the validation process.

Footnotes


4. The alternative to sampling is to review all of the records subject to validation.
5. TAR Glossary at 21.

6. TAR Glossary at 27 (each document in the sample should have an equal chance of being drawn).

7. See TAR Glossary at 13 (Documents above the Cutoff are Deemed to be Relevant and Documents below the Cutoff are deemed to be Non-Relevant.)

8. Reportedly, the court in *Global Aerospace, Inc. v. Landow Aviation, L.P., No. CL 61040 (Vir. Cir. Ct. Apr. 23, 2012)*, accepted a proposed standard of 75% recall at least in part on grounds that keyword search and manual review often achieved much lower recall. See Schieneman & Gricks at 264.

9. TAR Glossary at 25; Schieneman & Gricks at 273.

10. See e.g. What Should You Do With the Discard Pile?. and Your TAR Temperature is 98.6 – That's A Pretty Hot Result.

11. TAR Glossary at 27.


13. For example, the one at www.raosoft.com/samplesize.html

14. Courts have accepted samples having a confidence level of either 95% or 99%, and a nominal confidence interval of between ±2% and ±5% as satisfying the reasonable inquiry requirements of Rule 26(g). Schieneman & Gricks, at 270. The only change to the analysis resulting from different choices on confidence levels and confidence intervals is to change the required sample size.

15. The exact confidence interval is the range around the point estimate that could contain the true value of the feature being sampled. It is similar to, but not the same as, the margin of error that is used initially to determine your sample size. Once the sample is taken, we can calculate the exact confidence interval because it is dependent in part on the proportional results of the sample itself. You can read more about this at https://en.wikipedia.org/wiki/Binomial_proportion_confidence_interval.

17. Also known as prevalence or yield, richness refers to the number of relevant documents in a given population. TAR Glossary at 26.

18. E.g. Schieneman & Gricks at 273. “Since there are alternative means of calculating recall that do not require such a significant effort, whether this level of exactitude is required to satisfy the reasonable inquiry requirements of Rule 26(g) must be evaluated against the proportionality considerations in Rule 26(b)(2)(C)(iii).”


20. This is sometimes called the Basic Ratio method. See Grossman & Cormack Comments at 308, citing Schieneman & Gricks at 273.

21. This was reportedly used in the Global Aerospace case. See Grossman & Cormack Comments at 308 and Schieneman & Gricks at 273.


23. Noted blogger Ralph Losey points out this problem in his discussion of ei-Recall, which he proposes to use for recall validation in Introducing ‘ei-Recall’ – A New Gold Standard for Recall Calculations in Legal Search. Losey adds an “Accept on Zero Defect” process to his methodology along with a suggested stratified sampling approach for extremely low richness collections. See also Grossman & Cormack Comments at 308-310.

24. See Schieneman & Gricks at 264.

25. Compare Gricks & Schieneman at 273-274 (Since every step of the technology-assisted review process impacts either the nature or efficacy of the search, or the level of inquiry, Rule 26(g) applies throughout the process, from collection through validation. As with all discovery, what is reasonable in the application of Rule 26(g) to technology-assisted review is governed primarily by the proportionality considerations of Rule 26(b)(2)(C)(iii).
Simulations: How Could You Be So Sure?

Anyone who has watched the epic legal drama *My Cousin Vinny* realizes that the critical question in evaluating any claim is “How could you be so sure?” In our case, the answer is in large part: simulations.

Throughout this book we have made a number of claims and recommendations about the kinds of factors and workflows that should go into a state-of-the-art technology-assisted review (TAR) platform. These include things such as one-time versus continuous training (TAR 1.0 vs 2.0), who should do the training, what effect families have on overall review effectiveness and so on. How could we be so sure, you might ask?

For the past eight years, we have been running simulations to test every hypothesis, evaluate every factor and analyze every workflow relative to our continuous active learning (CAL) protocol that time, resources and imagination would permit. And to answer that same question for our clients, we have simulated numerous reviews to demonstrate the effectiveness of Insight Predict and how it compares with any other review protocols they may have used.
This chapter is devoted to an explanation of the method, measurement and value of running simulations to assess and improve technology-assisted review tools, and to compare the results of various review protocols inter se. The goal is to give you a basic understanding of the tool that should always be used to critically evaluate the claims of fact that abound in the marketplace—a well designed and properly executed simulation.

**How Does a Simulation Work?**

We apply the Cranfield model of information retrieval evaluation for simulations, which is a standard in the academic community. The Cranfield model essentially allows us to take existing data and simulate users interacting with Predict. In a simulation, we emulate a review just as it might happen in the field. When a human judgment is required, we enter that data programmatically just as a human reviewer might. Other than what is entered, the Predict algorithm has no other information about how the documents were tagged.

By running the simulation this way, we can automate and simulate the entire review in precisely the same order in which every document would have been reviewed in an actual Predict review. The difference is there is no need for human involvement, which makes it much quicker and easier to implement. It also allows us to run dozens of simulations, testing different methods or options using the same reviewer judgments each time. The essence of scientific method is that we keep all factors constant other than the one being tested.

We can thus run simulations to evaluate the impact of various factors upon the operation of Predict—things like the feature set, frequency of re-ranking, or document batching technique—or compare a Predict review to other review protocols. Every simulation follows the same general protocol shown in the following diagram.
To run the simulation, you need two things:

1. A collection of documents, and

2. A set of judgments on those documents.

For purposes of a Predict simulation, it is not essential to have the native version of every document. But you do need a document identifier (usually a Bates number, control number or document ID number), and the text of the document. You also need actual judgments on every document that is going to be evaluated in the simulation. Those judgments are often referred to as the “ground truth” for the simulation. We don't interpose or override reviewer judgments in our simulations.

Here Are the Steps We Follow:

1. **Create the Predict database**

   We begin by creating what is called a “graph” database covering all of the documents in the simulated review collection. Loosely speaking, the graph database analyzes the similarity in document text and creates a relationship map between and among every document in the collection and every word in those documents. That step is not shown in the diagram.
2. **Select initial training documents**

Then we select initial training documents, or seeds, to kick off the Predict ranking. We can, and have, selected those seeds in any number of ways, and in any quantity. We have used random seeds and judgmental seeds. We have even used synthetic seeds—manufactured documents that are not in the original collection, but include the type of language that we are looking for. And we have started with as few as a single document, or as many as 1,000 seeds to initiate the Predict ranking (although there is actually no limit).

Once we have selected the seeds, we apply the ground truth judgment (literally the reviewer's tag, relevant/non-relevant) for each document as if the documents were coded directly in the review tool. That is the simulation aspect of the Cranfield model. Those judgments initiate the Predict ranking, and the entire collection is ranked to push the best documents to the top of the list.

3. **Run the simulation to conclusion**

From that point forward, the process is pretty much automatic. A batch of documents is pulled from the top of the Predict ranking. That batch can be any size, but we often grab 50 document batches to roughly simulate an hourly review effort. We then apply ground truth to that batch of documents and evaluate our progress against our objectives. For most simulations, we rank the entire collection again, pull the next batch, and continue the process until we have simulated the review of every document in the collection.

**What Are the Results?**

What every Predict simulation provides us, more than anything else, is a document ranking/ordering. We know which documents were batched for review, in what order. So we also know the order in which the documents were coded with the ground truth.

Knowing the document ordering allows us to plot the progress and
results of the simulation on a yield curve, similar to the one in the figure below (from our family batching simulations, discussed in chapter 15). Knowing the ground truth and the document ordering, we can plot the percentage of the collection reviewed (or the number of documents, in this case), on the x-axis, and we can plot our progress on the y-axis.

In the example below, we are plotting the percentage of responsive documents found, essentially the overall recall of the review, which is the usual measure of progress.

Knowing the ground truth and the document ordering also provides us with a degree of flexibility in charting our progress and results. For example, when we wanted to evaluate the efficiency of the refined two-phase workflow that we evaluated along with family batching in chapter 15, we were able to assign a review rate to every ground truth decision that we applied. That allowed us to plot review time on the x-axis, rather than the percentage of documents reviewed. And, depending on what we are evaluating, we can plot different progress metrics on the y-axis. So, when we wanted to see how quickly Predict can elevate hot documents, or documents related to individual issues, as we discussed in dealing with investigations in chapter 7, we were able to plot the hot document recall and the aspectual recall, rather than overall recall on the basis of responsiveness.
Single Point Simulations

In some cases, we want to look more closely at just one specific point in the yield curve as our metric, rather than considering the yield curve as a whole. This is typically how we evaluate the impact of modifying different aspects of the overall Predict protocol, and it is a standard approach in information retrieval analysis. For example, there was a study done in 2017 to evaluate the relative effectiveness of using just the words in a document as the features for a TAR algorithm, versus using the concepts (latent semantic indexing, aka LSI) that those words represent across the entire collection of documents.\footnote{1} The metric they used to compare results was the precision of the review set once 75\% recall was achieved. Again, since the document ordering and ground truth are known in the simulation process, these types of metrics are easily derived.

Simulating Document Ordering

Finally, there are occasions when the document ordering itself is the actual result. For example, when we evaluated the effectiveness of Predict at finding hot documents that we discussed in chapter 15, the client wanted to know specifically which of the hot documents were the hardest to find. The yield curve told us how quickly we found the Hot documents, but the document ordering was needed to determine which were the most difficult—that is, the last to be reviewed and coded during the Predict review.

Overall, simulations provide us with a wealth of information that allows us to evaluate most any aspect of a Predict review.

Using Simulations to Compare Review Protocols

Over the years, we have run simulations to compare Predict with different review protocols. In most cases, clients ask how much Predict would have saved in comparison to a linear review of the entire collection. In other cases, we have compared Predict with other approaches, including keyword culling and other TAR methods (e.g. TAR 1.0).

As an example, we were asked to determine how much Predict would have saved against a linear review of a collection of just over 500,000
documents that was 41.9% rich (in other words, nearly 42% of the entire collection was responsive). The yield curve for that simulation appears below. The diagonal black line represents a random, linear review, and the blue curve reflects the results of the Predict simulation. The solid green line shows the number of documents that would have to be reviewed to find 80% of the responsive documents using a linear review (although, in reality, the entire collection is typically reviewed in a linear review). And the dashed green line shows how many documents had to be reviewed through Predict to achieve 80% recall.

You can readily see from the yield curve that the client could have saved on the review of roughly 156,000 documents by using Predict, which equates to a 37% cost savings, even compared with a linear review that unrealistically stops at 80% recall.

As another example, we were asked to evaluate the benefit of keyword culling before subjecting the collection to a Predict review. The small collection of about 4,900 documents had been culled to an even smaller collection of about 3,600 documents, with just under 700 responsive documents that had been linearly reviewed. To evaluate the benefit of culling, we separately simulated the review of the full collection and the culled collection. And we had to assume that the documents removed by culling were non-responsive.
The yield curves for both simulations appear together below. The red curve represents the Predict review of the entire collection, and the blue curve represents the Predict review of the culled collection. The solid green line shows the number of documents that would be reviewed to find 80% of the responsive documents if Predict were used on the entire collection, and the dashed green line represents the number of documents that would be reviewed to find the same number of documents by reviewing the culled collection with Predict. The difference is obvious, and we were able to advise the client on the limited benefit of culling.

Simulations like these are very helpful for clients as they consider alternative protocols for reviewing different document collections, because the simulations provide an objective basis for comparison and evaluation.

**Using Simulations to Improve Our Algorithm and TAR Processes**

From the beginning (seven years really), we have used simulations to evaluate the impact of various factors on the operation and efficiency of Predict. This has allowed us to refine our approach to the
continued development of our algorithm on any number of levels. It has also allowed us to optimize our Predict processes in order to make them as efficient as possible. You can see the results of these simulations in most of the chapters in this book.

One example of using a simulation to evaluate the impact of operational factors on the efficiency of Predict that we have not discussed elsewhere in this book is our analysis of “refresh rate” or the frequency of re-ranking the collection. To evaluate the impact, we simulated a Predict review of three different collections, and for each one, we tested three different refresh rates. We tested a weekly re-ranking and a daily re-ranking, because those rates are being used by other TAR tools. We also tested the impact of re-ranking once every 10 minutes. The yield curves for each of those three collections appear below (green is weekly, purple is daily, and light blue is every 10 minutes).

![Yield Curves for Three Refresh Rates](image)

This first collection shows that, for some collections, it simply doesn’t matter how often you rank. As you can see, the yield curves for each refresh rate are nearly overlapping. There may be some degradation in performance if you wait for an entire week to re-rank (the green curve), but the difference is slight.
In the second collection, the difference between refresh rates is more marked in every respect, particularly at typical recall rates of around 80%. Ranking every 10 minutes (light blue) will lead to a more efficient result than ranking every day (purple), and ranking every week will degrade performance even more (green).

In the third collection, the impact of refresh rate is worse yet. Re-ranking on a weekly basis (the green curve) seriously degrades
performance, to the point that recall rates barely change toward the end of the week, just before the ranking. Re-ranking on a daily basis (purple) is much better, but still does not compare with the Predict efficiency realized by re-ranking every 10 minutes.

These simulations lead to one conclusion: The rate of re-ranking does not always impact the efficiency of Predict, but when it does, the more often Predict re-ranks the collection, the better the results. Based on these simulations, we made sure that our Predict engine was designed to re-rank the collection as often as possible, at least several times every day, rather than daily or weekly.

**Conclusion**

The first question that lawyers should ask themselves when confronted with the question of whether a system or workflow or approach is good is, “Relative to what?” For example, in chapter 11 we made the claim that keyword searching is not a good approach to document review. However, that negative value assessment is due to keyword searching being compared against a well-executed supervised machine learning (TAR) approach. Relative to that TAR baseline, keyword search gets both lower recall and lower precision.

Once we have an answer to “Relative to what?” there is a second question to ask: “How do you know?” (How could you be so sure?). Lawyers often respond through logic and intuition but that is not enough for a data scientist. We know because we test using simulations. We can be so sure because we run our simulations over and over until we have a valid basis to be sure.

These Cranfield-style simulations are what allow us to know whether you can train with contract versus senior attorneys. Or whether CAL will get you high aspectual recall. Or whether you can start a TAR review with a thousand training documents, or only one training document. Or whether you should batch documents in review as families or not. Or whether re-ranking frequency and speed has any effect on the system. Or whether the core learning algorithm should use concepts (e.g. LSI) versus raw words as features. That last question is particularly pointed, because sometimes the outcome is
to not use something that seems intuitively like it should help, like concept search. It is counterintuitive that leaving concepts out of a TAR system actually yields better system, but that's something that comes about as a result of doing these simulations.

We have been running these kinds of simulations almost non-stop on all different kinds of questions for the past eight years. And the nice thing about simulation is that once you set up the parameters of the experiment, it'll run on its own, with no additional human involvement. So these experiments have been running over nights and weekends, over holidays, during vacation periods, during business trips, and so on. This constant experimentation, on real data, is what sets us apart.

Footnotes

1. This study used latent semantic indexing (LSI) to generate the concepts that would be used as features for comparison with the actual content of the document. Yang, et al., Effectiveness Results for Popular E-Discovery Algorithms, ICAIL 2017, London.

2. We have discussed several of our simulations in other chapters. In chapter 7, we looked at the use of random and synthetic seeds to initiate the ranking, as well as the ability to elevate Hot documents and documents relating to different issues in the case. In chapter 14, we compared the effectiveness of training with contract reviewers versus subject matter experts. And in chapter 15, we looked at the difference between batching by family as opposed to implementing a document-level review, and further evaluated the benefit of a refined two-phased workflow.
Case Study: Major Bank Slashes Review Costs with Innovative E-Discovery Technology

Insight Predict Cuts Production Review Costs by 94%

Our client was a large banking institution embroiled in nasty litigation with a now-defunct borrower. The bank alleged it lost millions due to the borrower’s principals’ accounting fraud. Legal shots were fired, excuses ran rampant and the parties went hard at each other to see where the blame would end up. Bring on the discovery.

The Problem

Responding to a production request, our client conducted an extensive investigation to find responsive documents. Even after using a variety of techniques to cull those that it found, it was still
left with over 2.1 million that needed consideration. Further keyword searching might have resulted in more reductions but the team wasn’t comfortable with what that process might miss.

Realizing they had neither the time nor money to review all 2.1 million documents, client and counsel turned to Insight Predict, Catalyst’s unique technology-assisted review (TAR) engine. The plan was to employ Predict’s continuous active learning (CAL) protocol and see if TAR might be effective in further reducing the population in a defensible manner.

**Step One: Building a Seed Set**

In this case, counsel had already reviewed and identified approximately 50,000 relevant documents for a previous production based on a similar request. Because our predictive ranking engine has no effective limit on the amount of training seeds it can handle, we used these documents as initial seeds to start the ranking process. Almost immediately, relevant documents from the larger collection were pushed to the front of the line for review.

**Step Two: Immediate Review**

Reviews using first-generation TAR 1.0 systems cannot begin until senior lawyers (aka subject matter experts) train the system. With Insight Predict’s advanced TAR 2.0 technology, this is not necessary. Rather, the review team could immediately begin requesting batches of documents to review.

Predict’s algorithm provided batches made up primarily of the documents it most highly ranked. This ensured that the review team was productive immediately, because they were focused on the documents that were most likely relevant. In turn, it enabled the trial team to quickly get their hands on the most important documents to help sharpen their analysis of the case.

The review batches also included a mix of documents chosen based on their “contextual diversity.” This unique feature of Insight is designed to solve the problem of “you don’t know what you don’t know.” Specifically, our contextual diversity algorithm chooses documents for review that are different than those already reviewed.
In effect, the algorithm clusters unseen documents by their common themes. It then pulls the most relevant examples from each cluster and presents them to reviewers as part of the batch. If the reviewer tags an example as relevant, the ranking engine is cued to promote similar documents. If the example is not relevant, the ranking engine learns that this cluster is of lesser interest.

The ultimate goal of the CAL protocol is to feed reviewer judgments back to the system to improve the training and thereby the responsiveness rate for subsequent review assignments. As the reviewers release their batches, Insight Predict adds their judgments to further its training. The net result is that the algorithm gets smarter and smarter about finding and promoting relevant documents. This is in sharp contrast to the “one-time” training used by the earlier TAR 1.0 systems.

**Step Three: Using Keyword Search to Further Improve the Training**

Predict’s ability to use flexible inputs allowed the review team to take a multimodal approach to finding responsive documents. As the review progressed, Predict identified promising search terms as well as custodians who held the most likely relevant documents.

This enabled the trial team independently to run searches with these key terms and then tag the relevant documents found through their searches.

As with the regular review, these tagged documents could then be fed into Predict’s ranking engine to further improve the training. This way, every attorney judgment on a document was used by Predict, no matter where that judgment was made.

**Step Four: Completing the Review**

Early on in the review, we took a richness sample to get a feel for the number of relevant documents to expect in the collection. That sample suggested we would find one relevant document out of every 100, which translates to a richness estimate of 1%.

As the review progressed, we tracked the number of relevant
documents found by the team. Toward the beginning, the team tagged 10% relevant, representing a ten-fold increase in review productivity. Over time, that figure rose to 25% and sometimes as high as 35%. Through their tagging, the review team showed that the predictive ranking process was paying dividends.

Eventually, the relevant documents petered out, dropping down to and below base richness. At this point the team decided to stop the review and measure their progress.

**Step Five: Measuring the Results**

To determine how many relevant documents the team had found, we ran a systematic random sample against the entire document population. In this case the team chose to sample just under 6,000 documents, which is larger than the typical discovery sample. The goal was to present results with a high level of confidence and a narrow margin of error. Senior attorney reviewers manually reviewed these documents for added credibility.
Conclusion: 98% Recall After Reviewing Just 6.4% of the Total Population

The sample suggested that the team had found and reviewed 98% of the documents relevant to the production. This conclusion was based on a sample confidence level of 95% and a 2% margin of error. Even taking the lower end of the margin-of-error range, we estimated that the team had found at least 92% of the relevant documents, still well beyond levels previously approved by the courts.

All of this was accomplished through a CAL workflow that put attorney reviewers’ eyes on every document produced, yet still required a total review effort of only 6.4% of the total reviewable population of 2.1 million documents. That was 1.97 million documents the attorneys were saved from having to review.
Case Study: Predict Proves Effective for Small Collection

Facing Tight Deadline in SEC Probe, Company Reduces Review by 75%

The question has persisted since technology-assisted review got its start. How big does a case need to be before it makes sense to load it into a TAR system? Are 10,000 documents enough? How about 100,000? In this case, it was just 16,000 documents, but TAR enabled the company to cut its review by 75% and get it done in under a week.

The Problem: A Tight Deadline and a Fixed Budget

Our client was a major law firm representing a company in an SEC investigation on a fixed-fee basis. It had a relatively small document set to review—just 16,000 documents. But the deadline was tight, the review had to be done quickly, and the company did not want to use contract attorneys, preferring instead to have the firm’s own lawyers take the laboring oar.
With So Small a Set, Would TAR Make Sense?

Mindful of its tight deadline, the firm wanted to use Insight Predict, our TAR 2.0 predictive ranking engine, to find the relevant documents more quickly. But it was concerned that using TAR with so few documents would be costly and ineffective.

With a first-generation TAR 1.0 system, the concern would have been justified. With TAR 1.0, they would need a senior lawyer to serve as subject matter expert to start the training process. The SME would first have to review and tag 500 or more randomly selected documents as a control set to use for measuring training efforts. Then the SME would have to do multiple rounds of training before the review could even begin.

This might require review of 1,600 or more documents before the system stabilized. Depending on the case, the training could require review of many more documents, easily as many as 3,000.

After that, the SME would have to tag an additional sample, perhaps another 500 documents, just to test whether the training was complete. All told, the SME might have to review 3,000 to 4,000 documents just to train the TAR 1.0 classifier. Only then could the review begin.

That's a lot of work for a case with a small number of documents. It is why many e-discovery professionals advise that TAR should only be used for larger cases. Some suggest the threshold is 100,000 documents before you can justify the SME's training efforts and expense.

Insight Predict Requires No Control Set or SME

With Insight Predict, the size of the collection is not a factor. With Predict's continuous active learning protocol, there is no minimum threshold because there is no need to create a control set. Predict ranks all of the documents all of the time. Predict does not require an SME to do the initial training. With no need for an SME, the review team can get started right away. Review is training and training is review.
In this case, the CAL process was quick and simple to implement. The team initially found 67 relevant documents through keyword search. We used these as training seeds for the initial ranking. Then the team started reviewing documents.

It turned out that about 11% of the documents were relevant, for a total of about 1,800 relevant documents. As the team reviewed documents, Predict continuously learned from their tagging and presented increasingly relevant batches to the reviewers. Relevance in the batches quickly rose to as high as 70%. Review efficiency increased seven-fold as a result.

In just days, batch relevance dropped to single digits as the team depleted the relevant population. The team stopped review and moved to a systematic sample to determine what level of recall was obtained. The sample involved 800 documents for a confidence level of 98% and a 4% margin of error. By this point, the team had reviewed just 3,900 documents.

The sample showed the team had found 96% of the relevant documents after reviewing only 25% of the population. The remaining documents could be safely discarded cutting out 75% of the review time and costs.

**Conclusion: Insight Predict is Effective Even for Small Collections**

Does TAR work for smaller cases? With Insight Predict and continuous active learning, the size of the collection does not matter. In this case, the team used Predict to finish their review in less than a week. They easily met the SEC’s deadline while also keeping well within their fixed-fee budget.
Case Studies: Chapter 24

24


Continuous Active Review Cuts Cost by Over 85%

Our client was a multinational Japanese company facing a large document production in an international patent dispute. The initial review collection exceeded 2 million documents. After a series of rolling uploads, which continued throughout the review, the population slated for review grew to 3.6 million. Facing millions in review costs, the client sought an alternative to linear review.

Review time was short. The client’s goal was to finish the review in four weeks with a small team handling the project. The documents were primarily in Japanese, with some English in the mix, and many involved highly technical subject matter.
Estimating Richness and Training

Even though the client had taken steps to remove junk and other documents not subject to production, the collection’s estimated richness was still miniscule. An initial systematic random sample of 1,000 documents (97% confidence with a 3.5% margin of error) suggested that there were fewer than six relevant documents in every thousand that might be presented through a linear review. As is often the case in litigation, richness was low at 0.6%.

Before Catalyst was engaged, a team of lawyers had reviewed about 10,000 documents found through keyword search. For many TAR 1.0 engines, which have a limited training phase, these judgments would have been of no use. Because Catalyst’s technology, Insight Predict, is a TAR 2.0 engine that uses continuous learning and continuous ranking, we could make use of these judgments as initial training seeds.

As you can see from the below yield curve, the initial training using the 10,000 seeds proved effective. It indicated that almost all of the relevant documents could be found after reviewing just 17% of the total review population. This meant that the review team could immediately exclude most of the non-relevant documents and start finding relevant documents many times faster than the day before. There was no need for the team to spend non-productive hours looking at largely irrelevant files selected randomly for initial training.
Optimizing Review with Continuous Active Learning

The initial training worked. Richness in the documents presented to the review team jumped from 0.6% to as much as 35%, which represented a 60-fold improvement in review efficiency. At the same time, the reviewers received a mix of documents selected for “contextual diversity.” This feature, integrated into Predict, allows the algorithm to keep finding and training against documents which are different from those already found through keyword search or seen by the reviewers in their initial rounds. You can read more about our unique contextual diversity algorithm on our website.

The review continued while the collection team added more documents. Since Predict can continually rank all the documents in the collection, there is no problem adding new documents during the review. As they are added, the documents are ranked and mixed into the total collection. To the extent they are similar to already ranked documents, they join the ranking in their proper place. To the extent they are different than what has already been collected, they become candidates for contextual diversity and can be included in the review sets for hands-on evaluation by the reviewers.

TAR 1.0 systems typically train against a reference set, which makes handling rolling collections difficult. To be representative, the reference set must be chosen randomly from the entire population and then carefully tagged by a subject matter expert. Then the training process begins with each round being measured by its effectiveness against the reference set.

If new documents are collected during the TAR 1.0 process, you have two options, with neither being ideal. Either you hope/assume that the new documents are similar to those already collected. Or you start again, discarding the initial reference set and its related training for a new round.

Rolling Collections and Continuous Learning

As mentioned earlier, through rolling collections over the course of several weeks, the Predict population grew to 3.6 million unique,
rankable documents. As the review team found new types of responsive documents and learned more about the case, they could also use any other search and analytics tools available to keep searching. Every decision they made was continuously fed back into Predict to improve its ranking. When the review team ran out of relevant documents, they stopped the review and conducted a further systematic random sample of the entire population. Here is what they learned:

As you can see from the resulting yield curve, Predict was still pushing relevant documents to the top of the review pile, even after multiple rolling collections were added while the review was in progress.

Ultimately, the total review effort was about 500,000 documents, out of 3.6 million scheduled for review. Predict allowed the review team to achieve the requisite recall after reviewing only a small fraction of the population, which met the client’s needs for both speed and efficiency. The team is now using Predict to help organize the review of all in-bound productions from other parties.

**Tokenizing Japanese Documents**

In the early days of technology-assisted review (TAR), many questioned whether it was suitable for Japanese and other Asian-
language documents. Indeed, for most TAR 1.0 engines, the answer was, and perhaps still is, a resounding “No.” After all, these products were designed to work on English-language documents that use spaces and punctuation to define word boundaries. For languages that do not follow Western syntax, the systems could not build the indexes required for them to work.

It is important to note that TAR systems don’t actually understand the words they index and analyze. Rather they employ mathematical algorithms to determine the frequency and use of the words both in the documents and across the document population.

Japanese and other languages that do not use spaces between words often have to be “tokenized” (broken out into artificial words) before they can be indexed for search and analytics tools. Many earlier tools do this in a simple way, just taking two or three characters at a time. While this approach works okay for basic search, it can make analysis very difficult for TAR 1.0 systems.

TAR 2.0 systems such as Insight Predict employ special software to tokenize Japanese and similar languages a smarter way. They are able to analyze the text and break out actual words and word phrases, not just arbitrary groups of characters. Once the Japanese documents were properly tokenized, the TAR 2.0 process could index and analyze them more effectively.

**Conclusion**

This case presented a number of challenges. The collection was mostly in Japanese and contained a number of highly technical documents. The richness of the collection was low and it contained a lot of junk. The client was on a tight timeline for review but collections kept arriving on a rolling basis.

Despite these challenges, we were able to make use of the 10,000 documents the legal team had already reviewed to jumpstart the ranking process and accelerate the review. Even with the collection’s low richness, the team was able to find highly relevant documents many times faster than with any other approach. And because Predict never stopped learning from newly-reviewed documents, it
continued to improve and help attorneys explore the collection even as new documents were constantly being added.

In the end, using Predict and its ability to support continuous active learning, the client was able to cut the time and cost of its review by over 85%.

**Postscript: A Supplemental Review**

Because of continuing collections in the case, another 132,000 documents were loaded after we finished the final systematic random sample. But because we could use the existing judgments and Predict training, our client was able to get through that last batch of stragglers quickly.

The review of that late-arriving batch had to have a separate validation procedure performed to support the decision to stop the review when the review team no longer saw relevant documents.

The procedure showed that the estimated recall on that last batch was 87%, even higher than the recall for the much larger, master collection that had accumulated over the rest of the review.
Case Study: Using Insight Predict for Review of Rolling Productions

*Insight Predict Finds 75% of Hot Docs While Cutting Review 92%*

Finding “hot” documents in an opposing party's production is rarely easy. But when those productions are large and arrive on a rolling basis, the search can be even more cumbersome, costly and time-consuming.

This was the scenario faced by plaintiffs in a lawsuit alleging predatory home-lending practices by a major financial institution. However, through the use of Insight Predict, the only technology-assisted review platform on the market that uses continuous active learning (CAL), coupled with Catalyst’s unique contextual diversity sampling, the plaintiffs were able to reduce the number of documents they had to review by 92%.
Challenge: Find Hot Documents in Opponent’s Rolling Productions

The plaintiffs in this case were working with limited resources to take on a major financial institution. In response to the plaintiffs’ discovery requests, the defendant had started to produce large numbers of electronic documents, with the productions arriving in waves on a rolling basis.

To prepare for depositions and further litigation, the plaintiffs had to quickly find the hot documents within these productions. But with limited resources, they could not afford to review them all manually.

Solution: Use Insight Predict to Create Prioritized Review

Two features of Insight Predict made it ideally suited to this case. First was continuous active learning, which gives it the ability to handle rolling productions. Because Predict ranks every document every time, new documents can be added continuously. This differs from earlier TAR platforms, which train against a small reference set and are therefore limited in their ability to handle rolling uploads.

Second, Predict differs from other platforms in its ability to effectively handle document populations with low richness (a low prevalence of relevant documents). In this case, when we evaluated the initial population of the defendant’s produced documents, we estimated that only about 1% were hot. For other platforms, that would have been a problem.

By using Insight Predict to rank the documents most likely to be hot, we were able to bring a higher concentration of them to the front of the review queue. Then, using Predict’s automated workflow, we fed these ranked documents to the review attorneys. Reviewers coded documents in small batches of 20, in order to take maximum advantage of Predict’s seamless continuous active learning. Each completed batch triggered new ranking rounds in the background (each running in under 10 minutes), such that dozens of rounds were run every day to integrate new review feedback and improve the next batches of documents served on-demand to the review team.
For the batches being fed to the reviewers, Predict quickly raised the richness of hot documents from 1% to 7%. That meant that the reviewers were getting seven times the richness they would otherwise have seen.

It also meant that they were able to find the majority of hot documents after reviewing only 8% of the collection. To understand this, compare these two graphs. The first shows the hot documents distributed randomly throughout the population:

This second graph shows the hot documents as ranked by Predict. The area shaded grey represents the last point we measured during this review. At that point, the attorneys had identified about 70% of the total predicted number of hot documents, but had reviewed only 8% of the produced population:

This flux curve further illustrates Predict’s ability to adjust to distinct events during the course of the review, such as the arrival of new productions and the arrival of new, untrained reviewers.

Contextual Diversity vs. ‘Hide the Ball’

One other feature of Predict that proved important in this case was its ability to perform contextual diversity sampling. Predict is the only TAR tool on the market with this ability. It samples the population to
ensure that there are no significant threads or pockets of documents that escape human review, even when a large proportion of the population will not have attorney eyes on it.

This has a significant benefit in a case such as this, where a plaintiff of limited means is up against a Goliath of a defendant. A common story in such cases has the defendant trying to bury revealing or damaging documents within a large, late production. When this happened during a traditional manual review, the documents might not have been noticed for some time.

However, with Predict's contextual diversity engine re-ranking and analyzing the entire document set every time, a pocket of new documents unlike anything reviewers have seen before is immediately recognized, and exemplars from those new pockets will be pulled as contextual diversity seeds and put in front of reviewers in the very next batch of documents to be reviewed.

**The Bottom Line**

These plaintiffs lacked the resources to engage in a brute-force review of the defendant's large, rolling productions. Insight Predict gave them the ability to quickly find the majority of hot documents and reduce the overall number of documents they had to review by more than 90%.
A Big Four accounting firm with offices in Tokyo asked Catalyst to demonstrate the effectiveness of Insight Predict, technology-assisted review (TAR) based on continuous active learning (CAL), for an investigation. They gave us a test population of approximately 5,000 documents which had already been tagged for relevance during a linear review. The firm had found only 55 relevant documents.

**Predict Simulation Overview**

Catalyst offered to run a free simulation designed to show how quickly Predict would have found those same relevant documents. The simulation would be blind—that is, Predict would not know how the documents were tagged until it presented its ranked list. That way we could simulate an actual Predict review using CAL.
We structured a simulated Predict review to be as realistic as possible, looking at the investigation from every conceivable angle. The results were outstanding so we ran it again, using a different starting seed. In fact, we did 57 different simulations starting with relevant seeds (singularly with each relevant document), a non-relevant seed and a synthetic seed.

Regardless of the starting point, Predict was able to locate 100% of the relevant documents after reviewing only a fraction of the collection. The numbers for achieving 80% recall were even better.

**Complicating Factors in the Document Collection**

Everything about the document collection in this investigation would normally be challenging for a TAR project.

**Japanese Language Documents**

To begin with, the entire collection was in Japanese. Like other Asian languages, Japanese documents require special attention for proper indexing, which is the first step in feature extraction for a technology-assisted review. At Catalyst, we incorporate semantic tokenization of the CJK languages directly into our indexing and feature extraction process, which is critical for a TAR project.

**Small and Low Richness Collection**

To complicate matters further, the collection itself was relatively small, and sparse. There were only 4,662 coded documents in the collection and, of those, only 55 total documents were considered responsive to the investigation. That put overall richness at only 1.2%.

The following example illustrates why richness and collection size together compound the difficulty of a project. Imagine a collection of 100,000 documents that is 10% rich. That means that there are 10,000 responsive documents. That’s a large enough set that a TAR engine will likely do a good job finding most of those 10,000 documents.

Next, imagine another collection of 1 million documents that is 1%
rich. That means that there are also 10,000 responsive documents. That is still a sizeable enough set of responsive documents to be able to train and use TAR, even though richness is only 1%.

Now, however, imagine a collection of only 100 documents that is 1% rich. That means that only one document is responsive, which means that either you’ve found it, or you haven’t. There are no other responsive documents other than that document itself, so there are no other documents that, through training of a machine learning algorithm, can lead you to the one responsive document. Thus, a 1% rich million document collection is a very different creature than a 1% rich 100 document collection. These are extreme examples, but they illustrate the point that small collections are difficult and low richness collections are difficult, but small, low richness collections are extremely difficult.

Small collections like these are nearly impossible for traditional TAR systems because it is difficult to find seed documents for training. In contrast, Predict can start the training with the very first coded document. This means that Predict can quickly locate and prioritize responsive documents for review, even in small document sets with low richness.

OCR Documents

Compounding these constraints, nearly 20% (10 out of 55) of the responsive documents were hard copy Japanese documents that had to be OCR’d. Generally, it is difficult to effectively OCR Japanese script because of the size of the character set, the complexity of individual characters, and the similarities between the Kanji character structures. Poor OCR will impair feature extraction which will, in turn, diminish the value of a document for training purposes, making it much more difficult to find responsive documents, let alone find them all.

Simulation Protocol

To test Predict, we implemented a fairly standard simulation protocol that we used for NIST’s TREC program and often use to show clients see how well Predict might work on their projects. After making the text of the documents available to be ingested into Predict,
we simulated a Predict prioritized review using the existing coding judgments in a just in time manner, and we prepared a gain curve to show how quickly responsive documents could be located.

Since this collection was already loaded into Catalyst’s platform, Insight Discovery, we had everything we needed to get the simulation underway: document identification numbers (Bates numbers); extracted text and images for the OCR’d documents; and responsiveness judgments. Otherwise, the accounting firm simply could have provided that same information in a load file.

With the data loaded, we simulated different Predict reviews of the entire collection to see how quickly responsive documents would be located using different starting seeds.

Here is how the simulation worked:

1. In each experiment, we began by choosing a single seed document to initiate the Predict ranking, to which we applied the accounting firm’s responsiveness judgment. We then ranked the documents based on that single seed.³

2. Once the initial ranking was complete, we selected the top 20 documents for coding in ranked order (with their actual relevance judgments hidden from Predict).²

3. We next applied the proper responsiveness judgments to those 20 documents to simulate the review of a batch of documents, and then we submitted all of those coded documents to initiate another Predict ranking.
We continued this process until we had found all the responsive documents in the course of each review.

**First Simulation**

We used a relevant document to start the CAL process for our first simulation. In this case, we selected a relevant document randomly to be used as a starting seed. We then let Predict rank the remaining documents based on the initial seed and present the 20 highest-ranked documents for review. We gave Predict the tagged values (relevant or not) for these documents and ran a second ranking (now based on 21 seeds). We continued the process until we ran out of documents.

![Figure 1](image)

**Gain Curve: Uniformly Evaluating Results**

We used a gain curve to uniformly evaluate the results of the simulated reviews. A gain curve is helpful because it allows you to easily visualize the effectiveness of every review. On the horizontal x-axis, we plot the number of documents reviewed at every point in the simulation. On the vertical y-axis, we plot the number of documents coded as responsive at each of those points. The faster the gain curve rises, the better, because that means you are finding more responsive documents more quickly, and with less review effort.
The linear line across the diagonal shows how a linear review would work, with the review team finding 50% of the relevant documents after reviewing 50% of the total document population and 100% of the relevant documents after reviewing 100% of the total.

The red line in Figure 1 shows the results of the first simulation, using the single initial random seed as a starting point (compared to the black line, representing linear review). Predict quickly prioritized 33 responsive documents, achieving a 60% recall upon review of only 92 documents.

While Predict efficiency diminished somewhat as the responsive population was depleted, and the relative proportion of OCR documents was increasing, Predict was able to prioritize fully 100% of the responsive documents within the first 1,491 documents reviewed (32% of the entire collection). That represents a savings of 68% of the time and effort that would have been required for a linear review.

**Second Simulation**

The results from the first random seed looked so good that we decided to try a second random seed. Those results were just as good.

![Figure 2](image-url)
In Figure 2, the gray line reflects the results of the second simulation, starting with the second random seed. The Predict results were virtually indistinguishable through 55% recall, but were slightly less efficient at 60% recall (requiring the review of 168 documents). The overall Predict efficiency recovered almost completely, however, prioritizing 100% of the responsive documents within the first 1,507 documents (32.3%) reviewed in the collection—again, a savings of nearly 68% compared with linear review.

**Third Simulation**

The results from the first and second simulations were so effective that we decided to continue the simulation. In the next simulation, we wanted to see what would happen if we used a lower-ranked (i.e., more difficult for the algorithm to find) seed to start the process. To accomplish that, we chose the lowest-ranked relevant document found by Predict in the first two simulations as a starting seed. This turned out to be an OCR’d document (which was likely the most unique responsive document) to initiate the ranking. To our surprise, Predict was just about as effective starting with this lowly-ranked seed as it had been before.³

The yellow line in Figure 3 shows the results of starting with the last document located during the first two simulations. The impact of
starting with a document that, while responsive, differs significantly from most other responsive documents is obvious. After reviewing the first 72 documents prioritized by Predict, only one responsive document had been found. However, the ability of Predict to quickly recover efficiency when pockets of responsive documents are found is obvious as well. Recall reached 60% upon review of just 179 documents—only slightly more than what was required in the second simulation. And then the Predict efficiency surpassed both previous simulations, achieving 100% recall upon review of only 1,333 documents—28.6% of the collection, and a savings of 71.4% against a linear review.

**Fourth Simulation**

For the next simulation, we decided to use a random non-responsive document as the starting point. The results were just as good as the earlier rounds. Figure 4 illustrates these results.

![Figure 4](image)

**Fifth Simulation**

We decided to do one more simulation just to see what happened. For this final starting point, we created a synthetic responsive Japanese document. We composited five responsive documents
selected at random into a single synthetic seed, started there, and achieved much the same results.\textsuperscript{4}

Sixth through 57th Simulation

The consistency of these five results seemed really interesting so we ran simulations using every single responsive document in the collection as a starting point. Though not our plan at the outset, we ultimately simulated 57 Predict reviews across the collection, each from a different starting point (all 55 relevant documents, one non-relevant document, and one synthetic seed).

Figure 6 shows that the results from every simulated starting point were, for the most part, consistent. Regardless of the starting point, once Predict was able to locate a pocket of responsive documents, the gain curve jumped almost straight up until about 60% of the responsive documents had been located.

In every case, Predict was able to find every one of the responsive documents without having to review even one-third of the collection.
Figure 6 shows the results of all 57 simulations.

![Graph showing results of simulations](image)

**Figure 6**

In the appendix are the specifics of each simulation at recall levels of 60%, 80% and 100%.

As you can see, the overall results mirrored our earlier experiments, which makes a powerful statement about the ease of using a CAL process. Special search techniques and different training starts seemed to make very little difference in these experiments.

**Key Takeaways**

One of the primary benefits of a simulation as opposed to running CAL on a live matter is that one can vary and control every aspect of a review to see how the system and results change when the parameters of the review change. In this case, we varied the starting point, but kept every other aspect of the simulated review constant. That way, we could compare multiple simulations against each other and determine where there may be differences, and whether one approach is better than any other.

The important takeaway is the fact that the review order of these various experiments is exactly the same review order that the accounting firm would achieve, had they reviewed these documents in Predict, at a standard review rate of about one document per
minute, and made the exact same responsiveness decisions on the same documents.

*Averaged across all the simulations, Predict was able to find just over half of all responsive documents (50% recall) after reviewing only 89 documents (1.9% of the collection; 98.1% savings). Predict achieved 75% recall after reviewing only 534 documents (11.5% of the collection; 88.5% savings). And finally, Predict achieved an otherwise unheard of complete 100% recall on this collection after reviewing only 1,450 documents (31.1% of the collection; 68.9% savings).*

Furthermore, Predict is robust to differences in initial starting conditions. Some starting conditions are slightly better than others. In one case, we achieved 50% recall after only 65 documents (1.4% of the collection; 98.6% savings) whereas in another it took 163 documents to reach 50% recall (3.5% of the collection; 96.5% savings). However, the latter example achieved 100% recall after only 1,352 documents (29% of the collection; 71% savings), whereas the earlier example achieved 100% recall after 1,507 documents (32.3% of the collection; 67.7% savings).

Overall, the key is not to focus on minute differences, because all these results are within a relatively narrow performance range and follow the same general trend.

**Other key takeaways:**

1. Predict’s implementation of CAL works extremely well on low richness collections. Starting with only 55 relevant documents out of nearly 5,000 typically makes finding the next relevant document difficult, but Predict excelled with a low richness collection.

2. This case involved OCR’d documents. Some people have suggested that TAR might not work well with OCR’d text but that has not been our experience. Predict worked well with this population.

3. All documents were in Japanese. We have written about our success in ranking non-English documents but some have
expressed doubt. This study again illustrates the effectiveness of Predict's analytical tools when the documents are properly tokenized.

These experiments show that there are real, significant savings to using Predict, no matter the size, richness or language of the document collection.

Footnotes

1. Party and claim facts have been changed to preserve client confidentiality. Our goal is to show the power of Insight Predict and not to comment about the specifics of any matter.

2. We chose to initiate the ranking using a single document simply to see how well Predict would perform in this investigation from the absolute minimum starting point. In reality, a Predict simulation can use as many responsive and non-responsive documents as desired. In most cases, we use the same starting point (i.e., the exact same documents and judgments) used by the accounting firm to initiate the original review that is being simulated.

3. We chose to review 20 documents at a time because that is what we typically recommend for batch sizes in an investigation, to take maximum advantage of the ability of Predict to re-rank several times an hour.

4. It is interesting to note that Predict did not find relevant documents as quickly using a non-relevant starting seed, which isn't surprising. However, it caught up with the earlier simulation by the 70% mark and proved just as effective.

5. Compositing the text of five responsive documents into one is a reasonable experiment to run. But it's not what most people think of when they think synthetic seed. They imagine some lawyer crafting verbiage him- or herself, writing something up about what they expect to find, in their own words. And then using that document to start the training. Using the literal text of five documents already deemed to be responsive is not the same thing but it made for an interesting experiment.
## Appendix: Simulation Results

<table>
<thead>
<tr>
<th>DocID</th>
<th>60%</th>
<th>80%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>27096</td>
<td>4%</td>
<td>15%</td>
<td>29%</td>
</tr>
<tr>
<td>34000</td>
<td>2%</td>
<td>11%</td>
<td>32%</td>
</tr>
<tr>
<td>35004</td>
<td>4%</td>
<td>12%</td>
<td>32%</td>
</tr>
<tr>
<td>83204</td>
<td>3%</td>
<td>11%</td>
<td>32%</td>
</tr>
<tr>
<td>86395</td>
<td>4%</td>
<td>14%</td>
<td>32%</td>
</tr>
<tr>
<td>93664</td>
<td>2%</td>
<td>13%</td>
<td>32%</td>
</tr>
<tr>
<td>98263</td>
<td>3%</td>
<td>11%</td>
<td>29%</td>
</tr>
<tr>
<td>98391</td>
<td>2%</td>
<td>13%</td>
<td>32%</td>
</tr>
<tr>
<td>98945</td>
<td>3%</td>
<td>11%</td>
<td>32%</td>
</tr>
<tr>
<td>99708</td>
<td>4%</td>
<td>12%</td>
<td>32%</td>
</tr>
<tr>
<td>99773</td>
<td>2%</td>
<td>10%</td>
<td>32%</td>
</tr>
<tr>
<td>99812</td>
<td>2%</td>
<td>11%</td>
<td>32%</td>
</tr>
<tr>
<td>99883</td>
<td>2%</td>
<td>12%</td>
<td>32%</td>
</tr>
<tr>
<td>100443</td>
<td>4%</td>
<td>12%</td>
<td>32%</td>
</tr>
<tr>
<td>100876</td>
<td>3%</td>
<td>13%</td>
<td>32%</td>
</tr>
<tr>
<td>101211</td>
<td>4%</td>
<td>12%</td>
<td>32%</td>
</tr>
<tr>
<td>101705</td>
<td>3%</td>
<td>14%</td>
<td>31%</td>
</tr>
<tr>
<td>101829</td>
<td>3%</td>
<td>11%</td>
<td>31%</td>
</tr>
<tr>
<td>102395</td>
<td>3%</td>
<td>13%</td>
<td>32%</td>
</tr>
<tr>
<td>102432</td>
<td>4%</td>
<td>14%</td>
<td>32%</td>
</tr>
<tr>
<td>102499</td>
<td>2%</td>
<td>9%</td>
<td>32%</td>
</tr>
<tr>
<td>102705</td>
<td>3%</td>
<td>14%</td>
<td>32%</td>
</tr>
<tr>
<td>103803</td>
<td>4%</td>
<td>12%</td>
<td>32%</td>
</tr>
<tr>
<td>105017</td>
<td>2%</td>
<td>14%</td>
<td>32%</td>
</tr>
<tr>
<td>105799</td>
<td>3%</td>
<td>13%</td>
<td>32%</td>
</tr>
<tr>
<td>106993</td>
<td>2%</td>
<td>12%</td>
<td>30%</td>
</tr>
<tr>
<td>107315</td>
<td>2%</td>
<td>14%</td>
<td>32%</td>
</tr>
<tr>
<td>109883</td>
<td>4%</td>
<td>12%</td>
<td>32%</td>
</tr>
<tr>
<td>110350</td>
<td>3%</td>
<td>15%</td>
<td>30%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DocID</th>
<th>60%</th>
<th>80%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>112905</td>
<td>4%</td>
<td>14%</td>
<td>32%</td>
</tr>
<tr>
<td>117037</td>
<td>4%</td>
<td>12%</td>
<td>32%</td>
</tr>
<tr>
<td>118353</td>
<td>4%</td>
<td>14%</td>
<td>32%</td>
</tr>
<tr>
<td>119216</td>
<td>4%</td>
<td>15%</td>
<td>32%</td>
</tr>
<tr>
<td>119258</td>
<td>2%</td>
<td>12%</td>
<td>32%</td>
</tr>
<tr>
<td>119362</td>
<td>2%</td>
<td>10%</td>
<td>32%</td>
</tr>
<tr>
<td>121859</td>
<td>3%</td>
<td>11%</td>
<td>32%</td>
</tr>
<tr>
<td>122000</td>
<td>4%</td>
<td>15%</td>
<td>29%</td>
</tr>
<tr>
<td>122380</td>
<td>5%</td>
<td>11%</td>
<td>30%</td>
</tr>
<tr>
<td>123626</td>
<td>3%</td>
<td>10%</td>
<td>32%</td>
</tr>
<tr>
<td>123887</td>
<td>3%</td>
<td>11%</td>
<td>32%</td>
</tr>
<tr>
<td>124517</td>
<td>3%</td>
<td>14%</td>
<td>32%</td>
</tr>
<tr>
<td>125901</td>
<td>3%</td>
<td>14%</td>
<td>32%</td>
</tr>
<tr>
<td>130558</td>
<td>2%</td>
<td>14%</td>
<td>32%</td>
</tr>
<tr>
<td>131255</td>
<td>4%</td>
<td>10%</td>
<td>32%</td>
</tr>
<tr>
<td>132604</td>
<td>2%</td>
<td>10%</td>
<td>32%</td>
</tr>
<tr>
<td>136819</td>
<td>3%</td>
<td>14%</td>
<td>29%</td>
</tr>
<tr>
<td>140265</td>
<td>4%</td>
<td>13%</td>
<td>32%</td>
</tr>
<tr>
<td>140543</td>
<td>4%</td>
<td>12%</td>
<td>32%</td>
</tr>
<tr>
<td>147820</td>
<td>3%</td>
<td>14%</td>
<td>32%</td>
</tr>
<tr>
<td>154413</td>
<td>4%</td>
<td>13%</td>
<td>32%</td>
</tr>
<tr>
<td>238202</td>
<td>4%</td>
<td>12%</td>
<td>32%</td>
</tr>
<tr>
<td>242068</td>
<td>4%</td>
<td>12%</td>
<td>32%</td>
</tr>
<tr>
<td>245309</td>
<td>4%</td>
<td>16%</td>
<td>32%</td>
</tr>
<tr>
<td>248571</td>
<td>4%</td>
<td>12%</td>
<td>32%</td>
</tr>
<tr>
<td>NR</td>
<td>3%</td>
<td>14%</td>
<td>32%</td>
</tr>
<tr>
<td>SS</td>
<td>2%</td>
<td>13%</td>
<td>31%</td>
</tr>
<tr>
<td>Min</td>
<td>2%</td>
<td>9%</td>
<td>29%</td>
</tr>
<tr>
<td>Max</td>
<td>5%</td>
<td>16%</td>
<td>32%</td>
</tr>
<tr>
<td>Avg</td>
<td>3%</td>
<td>13%</td>
<td>32%</td>
</tr>
</tbody>
</table>
Case Study: Predict Proves Effective Even With High Richness Collection

Finds 94% of the Relevant Documents Despite Review Criteria Changes

Our client, a global corporation, was hit with a federal investigation into alleged price fixing. The regulators believed they would find the evidence in the documents. The request to produce was broad, even for this three-letter agency. Our client would have to review over 2 million documents. And the deadline to respond was short, just four months to get the job done.

This wasn’t a case of finding a needle in a haystack. Rather, a wide range of documents were responsive. A sample of the initial collection suggested that as many as 45% of the documents would be responsive. One option was to produce everything but the client had a lot of confidential and proprietary information that it didn’t want
in the hands of competitors or the public. The assignment was to produce responsive documents but only responsive documents.

**Making Review Efficient**

Our goal was simple. Use Insight Predict, our TAR 2.0 continuous active learning (CAL) algorithm, to find the relevant documents as quickly and efficiently as possible. We matched our engine with the Catalyst Review Team, a bunch of well-trained Predict ninjas who are versed in getting the most out of our software.

We started using CAL from the beginning, no waiting around for a senior lawyer “subject matter expert” (SME) to wave hands of the document under the guise of training. Rather, the team got started right away using the responsive documents already identified for initial training.

The measure of a predictive review is how quickly the algorithm can surface relevant documents. Like a bloodhound born to track, Insight Predict picked up the trail almost immediately.

Here is a chart showing the percentage of relevant documents Predict found on a batch by batch basis.
There were almost 5,000 batches in this review. Each blue line represented the percentage of relevant documents in the batch. In the early stages, the number reached 80% to 90%. That meant the reviewer found 80 to 90 responsive documents out of 100 in their batch. It also meant the reviewer saw only a few nonresponsive documents, which was our ultimate goal. Make the reviewers as efficient as possible, to keep review costs as low as possible.

**Review Efficiency**

This is an important but seldom-discussed topic. How many nonresponsive documents does the reviewer see for each responsive one?

When keywords are used to cull documents, reviewers typically have to look at as many as nine nonresponsive documents for each responsive one. That means they are wasting their time for about 90% of their review efforts. With Insight Predict, our statistics show that the ratio is much narrower, about 2 to 1, which means the team finishes faster and bills less. We call this “review efficiency,” and it is an important ratio to consider when looking at e-discovery alternatives.

In this case, the team achieved a review efficiency of 1.33 to 1, which is pretty remarkable. That means that the review team looked at very few nonresponsive documents over the course of the project. Taking the math up two decimals, the average team member only had to look at 133 documents to find 100 responsive ones. Not a lot of time wasted.

That’s why the chart showed batches that quickly approached 100% responsive. Interestingly, you will see a responsive rate dip early on in the project. That was because the team had to finish a couple key custodians first and when they started running out of relevant documents, the numbers dipped. When we opened the review back up to the entire collection, the responsive rate jumped as well. Some batches were 100% responsive.

You can also see that batch richness dropped at the end. This is to be expected. With CAL, the goal is to keep reviewing until you stop seeing responsive documents. Once the responsive rate drops
substantially (say to a tenth of the high water mark), that is a signal that it is time to stop. In this case the team kept reviewing batches to make sure they were nearing the end.

**Adding Documents at the End**

A few spikes at the end of the process occurred because the team collected some additional documents and added them to the review. With Predict, rolling collections are not a problem. The added documents simply join the ranking and are promoted accordingly. That is what happened here. With TAR 1.0, in contrast, you have to start the training over again.

**Changing Review Criteria**

There was another wrinkle in the review process, although it is not uncommon. At a couple of points along the way, team leaders refined their view of responsiveness. This is a natural process as you learn more about your documents and about your case. Many call it relevance drift. The simple fact is that you know more about your needs at the end of the process than at the beginning. It is one of the biggest weaknesses of the old TAR 1.0 process. If all your training is done at the beginning, how do you account for what gets learned as you go along?

Catalyst's TAR 2.0 algorithm is noise tolerant, which means that coder inconsistencies and even changes in direction do not adversely affect it. With CAL, every ranking starts fresh, with no memory of the previous one. That way, if you were to retag tens of thousands of documents, the next ranking would take it in stride. The same is true as the team refines its search objectives.

That happened several times in this case as understanding increased. Yet, we could see no adverse impact on performance.

**The Results**

The team achieved 94% recall in this review, far greater than that required by the courts. They did so having reviewed only 60% of the collection. You can see how efficient Predict was in the adjacent chart.
This is a yield or gain curve. The x-axis shows the percentage of documents reviewed. The y-axis shows the percentage of responsive documents found.

The diagonal dashed line shows the expected progress of a linear review. In a linear review, if you look at 20% of the documents, you will expect to see, on average, 20% of the relevant ones. At 50%, 50%. At 100%, 100%.

The blue line shows a perfect ranking. It represents how the review would go if the algorithm pushed all of the responsive documents to the front and the reviewers never had to look at a single nonresponsive document. Of course, this never happens.

The red line shows how this review went. Because richness was high (45%) you can’t expect the typical straight up rise you might see with lower richness (we have many examples including our TREC published work). Rather, the proper measure is how it compares to the alternative: linear review.

So, for example, in this case the team found 80% of the documents after reviewing just 45% of the population. To get to 80% in a linear review, you would have to review 80% of the documents, or an additional 370,000 documents. At an average of $1.50 per document reviewed (and QC'd), the client saved $550,000 dollars using a Predictive Review.
No algorithm can achieve perfection. The question to ask is “How close did we come? In this case, the answer is pretty close. As we mentioned before, the team achieved a review efficiency of 1.33 to 1. That translates to viewing about 60% of the population to get better than 90% recall, which you can see in the chart. The review was about as efficient as it can be.

TAR 1.0 systems generally can’t handle low richness collections, requiring users to cull the population using keywords or otherwise until richness rises to 15% or higher. TAR 2.0 systems excel at low richness, which makes them a safer bet for overall e-discovery. This case study also showed us that TAR 2.0, at least in the form of Insight Predict, handles high richness collections as well.

The Catalyst review ninjas met their deadline with room to spare. Review costs were a lot lower than they would have been with linear review or keyword search.

Footnotes

1. Party and claim facts have been changed to preserve client confidentiality. Our goal is to show the power of Insight Predict and not to comment about the specifics of any matter.
About OpenText Catalyst

OpenText™, The Information Company, is a market leader in Enterprise Information Management software and solutions, enabling Intelligent and Connected Enterprises by managing, leveraging, securing and gaining insight into enterprise information, on-premises or in the cloud.

OpenText™ Catalyst designs, hosts and supports the world's fastest and most powerful document repositories for large-scale discovery and regulatory compliance. For more than 20 years, global corporations and their counsel have relied on Catalyst to help reduce litigation costs and take control of complex legal matters.

To learn more about technology-assisted review and OpenText™ Catalyst, visit www.catalystsecure.com or call 877.557.4273

To download the digital edition of the book, visit www.catalystsecure.com/TARforSmartPeople
Technology-Assisted Review has been a game changer for e-discovery professionals, offering dramatic savings in both time and review costs for savvy clients and their legal counsel. This book confronts the difficult issues with the first generation of TAR applications, while showcasing the newer, more advanced protocols coming with TAR 2.0.

Praise for TAR for Smart People

“This book is superb, and I don’t apply that word lightly. Superb!”

Craig Ball
E-discovery consultant and author, *Ball in Your Court*

“I am happy to recommend John’s book to all legal professionals who want to learn more about predictive coding, and to take their search for evidence to the next level.”

Ralph Losey
Chair, e-Discovery Practice Group, Jackson Lewis; author *e-Discovery Team* Blog

“Finally, there is a book about TAR that is clear, incisive, and actually fun to read.”

John M. Facciola
United States Magistrate Judge for the District of Columbia (Retired)

“We are pleased to recognize Catalyst for embracing continuous active learning, which, according to our research, advances the state of the art in technology-assisted review.”

Maura R. Grossman & Gordon V. Cormack
Ms. Grossman and Dr. Cormack are professors in the School of Computer Science at the University of Waterloo, Ontario